

POSGRADOS

MAESTRÍA EN — ELECTRÓNICA Y AUTOMATIZACIÓN

RPC-SO-19-No.277-2018

OPCIÓN DE
TITULACIÓN:

PROYECTOS DE DESARROLLO

TEMA:

DESARROLLO DE UN CONTROLADOR ÓPTIMO LQR
UTILIZANDO HERRAMIENTAS IOT PARA UN SISTEMA
DE PRESIÓN CONSTANTE CONTROLADO
REMOTAMENTE

AUTOR:

VLADIMIR ANDRÉS CHALÁN PADILLA

DIRECTOR:

CARLOS GERMÁN PILLAJO ANGOS

QUITO - ECUADOR
2020

Autor:



Vladimir Andrés Chalán Padilla.

Ingeniero en Mecatrónica.

Candidato a Magíster en Electrónica y Automatización,
Mención en Informática Industrial por la Universidad
Politécnica Salesiana - Sede Quito.

vladchalanp@hotmail.com

Dirigido por:



Carlos Germán Pillaño Angos.

Ingeniero en Electrónica y Control.

Magister en Ingeniería.

cpillaño@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

©2020 Universidad Politécnica Salesiana.

QUITO – ECUADOR – SUDAMÉRICA

VLADIMIR ANDRES CHALAN PADILLA.

***DESARROLLO DE UN CONTROLADOR ÓPTIMO LQR
UTILIZANDO HERRAMIENTAS IOT PARA UN SISTEMA DE
PRESIÓN CONSTANTE CONTROLADO REMOTAMENTE***

Índice general

Índice de Figuras	5
Índice de Tablas	8
Abstract	9
1. Introducción	11
1.1. Descripción general del problema	12
1.2. Objetivos	12
1.2.1. Objetivo general	12
1.2.2. Objetivos específicos	12
1.3. Contribuciones	13
1.4. Organización del manuscrito	13
2. Marco Referencial	14
2.1. Estado del Arte	15
2.2. Formulación del Problema	20
3. Metodología y Diseño	21
3.1. Metodología	22
3.2. Requerimientos	22
3.3. Diseño general del sistema	23
3.4. Diseño mecánico	26
3.4.1. Selección de bomba	26
3.4.2. Diseño de tubería y montaje del sistema hidráulico . .	28
3.5. Diseño eléctrico	29
3.5.1. Selección del variador de frecuencia	29
3.5.2. Selección de la fuente de poder	30
3.5.3. Selección del controlador lógico programable	31
3.5.4. Selección del transductor de presión	33

3.5.5.	Selección de la tarjeta Raspberry Pi	34
3.5.6.	Diseño de planos eléctricos	35
3.6.	Diseño del sistema IoT	38
3.6.1.	Protocolo MQTT	38
3.6.2.	Implementación de broker MQTT	39
3.6.3.	Selección de plataforma IoT	42
3.6.4.	Diseño de la interfaz en el dashboard de Node-RED	43
3.6.5.	Diseño de interfaz en Blynk	44
3.6.6.	Configuración de Node-RED en Raspberry	46
3.6.7.	Configuración e implementación de enlace entre Node-RED y broker MQTT.	48
3.6.8.	Configuración e implementación de comunicación entre Node-RED y Blynk	49
3.7.	Programación del PLC	52
3.7.1.	Enlace entre PLC Siemens S7-1200 y Raspberry	52
3.7.2.	Algoritmo del PLC	55
3.8.	Diseño del algoritmo de control	57
3.8.1.	Modelo matemático del sistema	58
3.8.2.	Controlabilidad y observabilidad del sistema	60
3.8.3.	Diseño del controlador LQR	61
3.8.4.	Diseño del controlador LQI	62
3.9.	Modelamiento y análisis	63
3.9.1.	Simulación y análisis del controlador LQR	64
3.9.2.	Simulación y análisis del controlador LQI	65
3.9.3.	Implementación de los controladores LQR y LQI en Node-RED	66
3.10.	Integración de sistemas	67
3.11.	Verificación y validación	70
4.	Resultados y Conclusiones	72
4.1.	Resultados y análisis	73
4.2.	Conclusiones	78

Índice de Figuras

2.1. Principio de operación de un sistema automático de presión usando un variador de frecuencia [He et al., 2010].	16
2.2. Respuesta de 3 controladores en un sistema de control de presión [Panwar et al., 2018]	17
2.3. Arquitectura de un sistema IoT para el monitoreo de calidad del aire [Chanthakit and Rattanapoka, 2018]	19
3.1. Modelo en V para el diseño mecatrónico [Gausemeier and Moehring, 2002]	22
3.2. Diagrama de bloques general del sistema de presión constante	24
3.3. Arquitectura integral del sistema de presión constante a desarrollarse	25
3.4. Curva característica de la bomba Thebe	27
3.5. Curva característica de la bomba Ebara PARA 106T	27
3.6. Sistema hidráulico modelado en 3D	28
3.7. Diagrama P&ID del sistema	29
3.8. Variador de frecuencia Allen Bradley 22A-A4P5N104	31
3.9. Fuente de poder Delta DRP012V030W1AZ	32
3.10. PLC Siemens S7-1200 1212C AC-DC-RLY	33
3.11. Sensor de presión Pixsys Electronics ECT-NAT.	34
3.12. Raspberry Pi 3 B+.	34
3.13. Diagrama de conexiones del variador de frecuencia Allen Bradley 22A-A4P5N104.	35
3.14. Diagrama de conexión del PLC Siemens S7-1200 1212C AC-DC-RLY.	36
3.15. Plano eléctrico de fuerza del tablero de control.	36
3.16. Plano eléctrico de control del tablero de control.	37
3.17. Arquitectura de funcionamiento de MQTT.	39
3.18. Esquema para suscribir y publicar a broker MQTT del sistema.	39

3.19. Google Cloud Platform.	40
3.20. Instancia creada en Google Cloud Platform.	40
3.21. Lista de contenido de directorios en máquina virtual con Ubuntu.	41
3.22. Arranque de broker MQTT en Ubuntu.	41
3.23. Dashboard de EMQX.	42
3.24. Clientes en el broker MQTT.	42
3.25. Tópicos en el broker MQTT.	43
3.26. Inicio de Node-RED en computador con Windows.	44
3.27. Diseño de interfaz en el tablero de Node-RED.	45
3.28. Configuración de dashboard en Node-RED.	45
3.29. Diseño de interfaz en plataforma Blynk.	46
3.30. Comandos en terminal de Raspberry Pi para modificar dhcpd.	47
3.31. Arranque y ejecución de Node-RED en Raspberry Pi desde el terminal.	47
3.32. Ventana de programación de Node-RED en Raspberry.	48
3.33. Nodos para suscripción y publicación en MQTT.	49
3.34. Configuración del servidor a usarse en los nodos MQTT en Node-RED.	49
3.35. Configuración del nodo MQTT en Node-RED.	50
3.36. Configuración de conexión con servidor de Blynk Cloud.	50
3.37. Configuración de nodo de lectura de Blynk.	51
3.38. Configuración de nodo de escritura de Blynk.	51
3.39. Arreglo de nodos en Node-RED para envío y recepción de datos de plataforma Blynk.	51
3.40. Esquema de comunicaciones entre PLC S7-1200, Raspberry y broker MQTT en la nube.	52
3.41. Configuración general del nodo S7 en Node-RED.	53
3.42. Configuración de las variables del nodo S7 en Node-RED.	54
3.43. Equivalentes de variables del nodo S7 en Node-RED.	54
3.44. Configuración de un nodo de entrada/salida usando S7 en Node-RED.	55
3.45. Arreglo de nodos para lectura y escritura de una dirección del PLC Siemens.	55
3.46. Normalizado y escalado de entrada analógica en TIA Portal.	56
3.47. Selección de modos de operación del sistema.	56
3.48. Normalizado y escalado de salida analógica en TIA Portal.	57
3.49. Movimiento de datos a dirección de salida analógica del PLC.	57
3.50. Señal de entrada y de salida de la planta ante un escalón de 50 %.	58

3.51. Respuesta del modelo estimado versus el sistema original en System Identification de Matlab.	59
3.52. Función de transferencia en tiempo continuo del sistema.	59
3.53. Código para verificar la controlabilidad del sistema.	60
3.54. Código para verificar la observabilidad del sistema.	60
3.55. Código de implementación del controlador LQR.	62
3.56. Valores obtenidos de K en diseño de LQR en Matlab.	62
3.57. Código para el diseño de LQI en Matlab.	63
3.58. Constantes K para controlador LQI obtenidas usando Matlab.	63
3.59. Diagrama de bloques de controlador LQR en Simulink.	64
3.60. Respuesta del sistema con controlador LQR sin compensador a un escalón unitario.	64
3.61. Diagrama de bloques en Simulink para algoritmo LQR en tiempo discreto.	65
3.62. Respuesta del sistema con controlador LQR y compensador a 28PSI como punto de consigna.	66
3.63. Diagrama de bloques en Simulink para algoritmo LQI en tiempo discreto.	66
3.64. Respuesta del sistema con control LQI a 28PSI como punto de consigna.	67
3.65. Arreglo de nodos en Node-RED para controlador LQI y LQR.	67
3.66. Algoritmo de control en nodo de función en Node-RED usando JavaScript.	68
3.67. Tablero eléctrico de control del sistema.	69
3.68. Sistema hidráulico resultante.	69
3.69. Producto resultante de la integración de sistemas.	70
4.1. Verificación de conectividad de broker MQTT con clientes.	73
4.2. Validación del controlador LQR y de la interfaz IoT implementada en el sistema.	75
4.3. Respuesta del sistema con LQR a diferentes puntos de consigna.	75
4.4. Respuesta del sistema con LQI a diferentes puntos de consigna.	76
4.5. Validación 1 del controlador LQI implementado en el sistema.	76
4.6. Validación 2 del controlador LQI implementado en el sistema.	77
4.7. Validación 3 del controlador LQI implementado en el sistema ante perturbaciones.	78

Índice de Tablas

3.1. Características técnicas del variador de frecuencia WEG CFW100040SDPLZ.	30
3.2. Características técnicas del variador de frecuencia Allen Bradley 22A-A4P5N104.	31
3.3. Características técnicas del PLC Siemens S7-1200 1212C AC-DC-RLY.	32
3.4. Características técnicas del sensor Pixsys Electronics ECT-NAT.	33
3.5. Características técnicas de la tarjeta Raspberry Pi 3 B+.	34
4.1. Valores de voltaje y corriente obtenidos en el sistema.	73
4.2. Tabla comparativa de los datos de simulación y de la planta real con controlador LQR.	74
4.3. Tabla comparativa de los datos de simulación y de la planta real con controlador LQI.	77

Resumen

El presente trabajo presenta el diseño de un controlador óptimo LQR (Linear Quadratic Regulator por sus siglas en inglés) y un controlador óptimo LQI (Linear Quadratic Integrator por sus siglas en inglés) para un sistema de presión constante mediante el bombeo de agua, usando variador de frecuencia. De igual manera, se diseñó una aplicación IoT (Internet de las cosas) para computador usando protocolo MQTT (Message Queuing Telemetry Transport por sus siglas en inglés) mediante el enlace con un servidor tipo broker. Además, se implementó una aplicación para teléfonos inteligentes, desde cuyas aplicaciones se pudo monitorear las variables del proceso y a su vez controlar los modos de operación del sistema y establecer los puntos de consigna. Por otra parte, se diseñó y construyó el tablero de control, mismo que consta de un controlador lógico programable con entradas-salidas analógicas y digitales, un Raspberry, elementos de protección y un variador de frecuencia que comanda la velocidad del motor de la bomba, misma que forma parte del sistema hidráulico, el cual se compone de válvulas de retención, válvulas de paso, manómetro y transductor de presión. Así, los resultados del controlador óptimo LQI comparados con los del controlador LQR evidencian que, al tener una acción integral en sistemas no lineales, se logra obtener un seguimiento del punto de consigna con tiempos adecuados de estabilización. Adicionalmente, el uso de protocolos MQTT para sistemas IoT permiten que no exista una latencia significativa entre la comunicación de dos dispositivos. Por ende, se puede decir que el monitoreo y el control del sistema es en tiempo real, al igual que sus datos históricos. Por lo tanto, se concluye que el sistema desarrollado es escalable a procesos más complejos debido a su robustez en hardware y en el algoritmo de control. Además, cuenta con costos y herramientas accesibles para el consumidor, permitiendo tener un control óptimo junto con la posibilidad de monitorear y controlar dichos procesos de forma remota.

Palabras clave: LQI, LQR, internet de las cosas, sistema presión constante, control remoto, control óptimo, industria 4.0.

Abstract

The design of an optimal LQR (Linear Quadratic Regulator) controller and an optimal LQI ((Linear Quadratic Integrator) one for a constant pressure system by pumping water using a frequency inverter is presented. Likewise, an IoT (Internet of Things) application for a computer was designed by using the MQTT (Message Queuing Telemetry Transport) protocol which uses a link with a broker-type server. Additionally, an application for smartphones was implemented, from which it was possible to monitor the variables of the process and in turn, to control the operation modes of the system and to establish the set points. On the other hand, a control electrical cabinet was designed and built with a programmable logic controller with analog and digital input-outputs, a Raspberry, electrical protection elements and a variable frequency drive that controls the speed of the peripheral pump motor that is part of the hydraulic system made up by check valves, a manometer and a pressure transducer. Additionally, an LQI controller was design because the LQR controller was not be able to obtain a tracking of the set point with adequate stabilization times in this nonlinear system. Furthermore, the use of MQTT protocols for IoT systems allow not to have significant latency between the communication of two devices. Therefore, it can be said that the system monitoring and control is carried out in real time like as well as its historical data. Consequently, it is concluded that the developed system is scalable to more complex processes due to its hardware robustness and the control algorithm. In addition, its price and tools are accessible to the consumers, allowing them to have an optimal control of the processes that require it, along with the possibility to remotely monitor and control them from a computer or from a smartphone.

Keywords: LQI, LQR, internet of things, constant pressure system, remote control, optimal controllers, industry 4.0.

Capítulo 1

Introducción

El uso de controladores más robustos y eficientes es uno más de los retos al que los Ingenieros de Control se enfrentan diariamente en el control de procesos y sistemas actuales, es por esto que nuevos controladores basados en LQR (Linear Quadratic Regulator por sus siglas en inglés) [Bharali and Buragohain, 2017] han sido desarrollados para tener un control óptimo y una respuesta más eficiente. El uso de controladores óptimos en plantas de bombeo es requerido para otorgar una mejor respuesta ante perturbaciones, lo que prolonga la vida útil de las bombas y de los sistemas de tuberías al evitar golpes de arietes ocasionados por fluctuaciones considerables en las presiones y caudales de las mismas.

Con el paso de los años, sin importar el controlador que se utilice, la supervisión y control de ellos se lo hace desde paneles táctiles o desde sistemas SCADA (Supervisory Control and Data Acquisition por sus siglas en inglés). Sin embargo, actualmente, la tendencia es el término Industria 4.0, generado en Alemania en 2011, no es nada más que una “cuarta revolución industrial”, por lo que los líderes de la industria están siendo optimistas y están creando grandes expectativas [Bassi, 2017]. Últimamente, con el uso de dispositivos inteligentes en el diario vivir, se ha podido acceder al IoT (Internet of things por sus siglas en inglés) con el cual se puede conectar mediante internet desde cualquier sitio, a cualquier hora y hacia cualquier dispositivo [Park et al., 2014]. Por ello, el uso de dispositivos IoT en el campo industrial se está volviendo cada vez más común, de tal manera que los ingenieros de campo se han visto beneficiados debido a la facilidad de adquisición y control de datos del proceso de manera remota. [Aris et al., 2015]. Es por esta razón, que este trabajo busca aportar a la implementación de herramientas de Industria 4.0 junto con el diseño de controladores óptimos.

1.1. Descripción general del problema

En los últimos tiempos, los sistemas de control en Ingeniería han sido usados para varias aplicaciones en la industria. Sin embargo, el uso de algoritmos óptimos de control son cada vez más necesarios, debido a su eficiencia en cuanto a recursos computacionales demandados para su ejecución y en base a la respuesta del proceso. Desafortunadamente, uno de los limitantes para la implementación de estos controladores en el campo industrial ha sido la incapacidad de ejecución de algoritmos complejos sobre controladores lógicos programables que permitan tener aplicaciones robustas.

Adicionalmente, con la nueva tendencia de fábricas inteligentes y el uso e implementación de herramientas de industria 4.0, se requieren soluciones inteligentes de monitoreo y control de procesos de manera remota a un bajo coste de implementación y mantenimiento que permitan a la industria mejorar su productividad e incrementar su innovación tecnológica.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un controlador óptimo LQR, mediante el enlace de una tarjeta Raspberry Pi3 y un PLC Siemens S7-1200, para aplicarlo en un sistema de presión constante controlado inalámbricamente de forma remota utilizando herramientas IoT.

1.2.2. Objetivos específicos

- Estudiar la implementación en sistemas de presión constante de controladores óptimos mediante el uso de la tarjeta Raspberry Pi3 y el PLC Siemens S7-1200, así como su modelamiento y control inalámbrico.
- Diseñar el controlador óptimo LQR, a través de la retroalimentación por transmisor de presión de línea de descarga de la bomba para la actuación sobre la velocidad del impulsor.
- Diseñar la interfaz de visualización y control remoto mediante la plataforma de IoT NodeRED, para la configuración de los elementos que comandarán el sistema.
- Realizar las pruebas de funcionamiento y control a través de la lectura remota de las entradas físicas y analógicas, salidas digitales

y analógicas del sistema, para determinar la eficiencia de la respuesta ante perturbaciones.

1.3. Contribuciones

El presente trabajo promueve la implementación de algoritmos de control óptimos en procesos sencillos y complejos junto con la integración de sistemas de Industria 4.0 a un coste accesible para el consumidor, contribuyendo al desarrollo de la cuarta revolución industrial y al crecimiento de la productividad del país.

1.4. Organización del manuscrito

El manuscrito se compone de 4 capítulos: introducción, marco referencial, metodología y diseño, resultados y conclusiones.

Capítulo 2

Marco Referencial

El capítulo 2 constará del estado del arte, donde se analizarán los actuales algoritmos de control para sistemas de presión así como también de los equipos y herramientas del internet de las cosas para el control de dispositivos de manera remota. Adicionalmente, se desarrollará la formulación del problema.

2.1. Estado del Arte

En la industria, para el control de motores y bombas se usan variadores de frecuencia los cuales son dispositivos que varían la velocidad de un motor de inducción asíncrono, mediante la modificación en la frecuencia que genera una corriente alterna con una frecuencia y voltaje necesarios para que el motor se pueda accionar independientemente de la frecuencia de la red. Así, una de las principales ventajas de un variador de frecuencia es que brinda ahorro energético, puesto que los picos de corriente de arranque disminuyen notablemente al tener una rampa de aceleración [Álvarez, 2000]. La implementación de motores eléctricos con variadores de frecuencia está cada vez más disponible en los últimos años y son estos los componentes clave en la disminución de potencia en sistemas de bombeo por motores [Rhyn, 2018]. De igual manera, el autor [Kato et al., 2019] concluye que en su caso de estudio, el uso de variadores de frecuencia reducen el consumo energético en un 10 %, sin embargo, puesto que en plantas de tratamiento de aguas residuales muchas veces se tienen diferentes puntos de operación, el algoritmo de control incorporado con el variador de frecuencia no es efectivo y requiere de algoritmos de control robustos.

El autor [Li and Liu, 2009] desarrolló un sistema automático de presión constante en lazo cerrado que usa un PLC (Programmable Logic Controller) S7-300, un SCADA (Supervisory Control and Data Acquisition) y que enlaza variadores de frecuencia marca ABB. Por otro lado el autor [Zhang and Zhuan, 2019] analiza la implementación de tres variadores de frecuencia con diferentes modos de operación de velocidad para la tercera estación de bombeo de la ciudad de Huaiyin en China y concluye que la eficiencia de costos depende directamente de la operación por variador de frecuencia.

Precisamente, las plantas industriales de bombeo por lo general están interconectadas y trabajando a presiones diferentes y por esta razón, el control de presión es un aspecto importante tanto en plantas eléctricas, industria química, perforaciones, automotores y estaciones de bombeo, entre otras áreas. Adicionalmente, el autor [Itani et al., 2020] menciona que los procesos de control de presión están usualmente expuestos a una frecuente variación en la carga, lo cual hace necesario que un control eficiente se lleve a cabo para evitar problemas de operación y daños al sistema hidráulico. Hace algún tiempo, los controles convencionales como PI (Proporcional-Integral) y PID eran usados en las industrias para regular la presión logrando un desempeño eficaz de controles convencionales para procesos simples con un rango específico de salida, pero no para cambios frecuentes en el punto de consigna ni para perturbaciones severas [Al-dhaifalla, 2017].

La figura 2.1 muestra el principio de operación de un control de presión automático usando un variador de frecuencia para sistemas de irrigación.

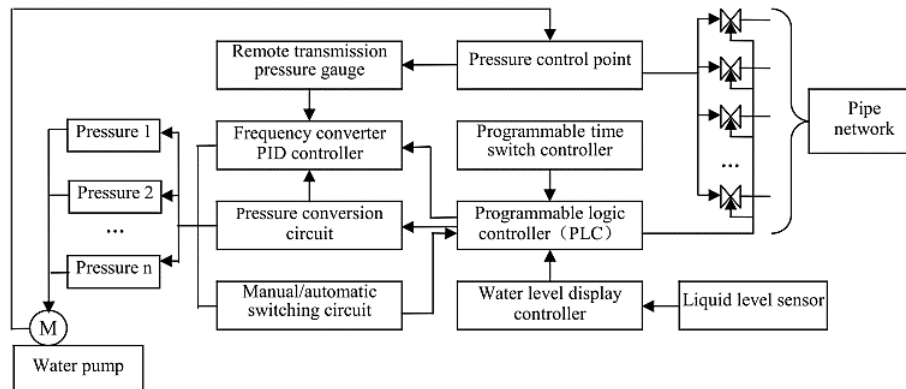


Figura 2.1: Principio de operación de un sistema automático de presión usando un variador de frecuencia [He et al., 2010].

Hoy en día, existe una rama de la teoría de control moderno que es el control óptimo, el cual resuelve problemas para sistemas dinámicos, haciéndolo diferente de los demás tipos de controles ya que encuentra una manera adecuada para identificar la ley de control que permite obtener un resultado óptimo. Este tipo de control consiste en un conjunto de ecuaciones diferenciales que definen el comportamiento de las ecuaciones de las variables de control que reducen la función de costo la cual es una función que incluye variables de control y de estado. Un ejemplo de estos controladores es el LQR

(Linear Quadratic Regulator por sus siglas en inglés) [Abdurraqeeb et al., 2019].

Por otra parte, los autores [Ozana et al., 2012], desarrollaron un controlador LQR para un péndulo invertido usando el filtro de Kalman como observador para un sistema de dos estados no medidos. El modelo matemático de este sistema considera el coeficiente de fricción y este sistema es implementado sobre un control REX aplicado en un controlador programable automático de marca WinPAC, de manera que obtienen una buena relación costo/rendimiento.

Además, los autores [Panwar et al., 2018] diseñaron un controlador PI, PI desacoplado y un controlador óptimo LQR para el control de nivel y presión en un tambor de presión de un reactor crítico avanzado de agua. La figura 2.2 muestra la clara comparación entre la respuesta de los tres controladores y claramente se observa que la respuesta del LQR es más rápida y no oscilante, logrando estabilizar la presión del sistema y el nivel del tanque en menos de 1 segundo en un rango amplio de operación.

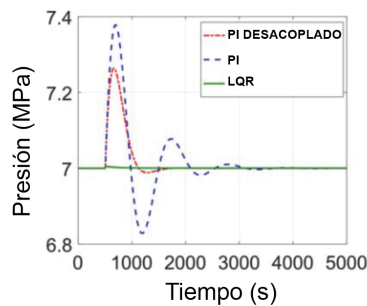


Figura 2.2: Respuesta de 3 controladores en un sistema de control de presión [Panwar et al., 2018]

Con el fin de lograr compatibilidad entre los PLCs estándar y la industria 4.0 se requieren equipos adicionales que muchas veces son costosos, por esta razón surge la idea de implementar tarjetas Arduino y Raspberry-Pi como una solución conveniente para adicionar capacidades de tecnologías web, procesamiento de datos por red y comunicaciones [Minchala et al., 2020]. Es el caso presentado en [Peng et al., 2020] donde se considera que la mayor parte de los PLCs en la industria tienen una capacidad limitada de memoria para almacenar registro de datos, en el caso de implementar tecnologías de Industria 4.0 estos datos pueden ser almacenados en la nube y así generar una eficiencia global en los sistemas automáticos.

Adicionalmente, uno de los conceptos tecnológicos principales en la Industria 4.0 es el internet de las cosas (IoT), el cual estipula que los objetos físicos como sistemas embebidos y sistemas cyber-físicos que contienen software deben estar conectados a una red (conectividad a internet) y típicamente a una aplicación de la nube, de manera que pueden crear oportunidades para nuevos servicios con la integración del mundo físico y el digital, por ejemplo, el monitoreo automático. En este caso, las “cosas” pueden ser varios ítems, por ejemplo, una banda transportadora en una fábrica. Consecuentemente, el término Internet Industrial de las cosas (IIoT) es el más adecuado al referirse a Industria 4.0. Así, los actuadores, sensores o sub sistemas pueden ser dispositivos IoT, mientras que la actividad de enviar datos a la nube, o actuar sobre un proceso a partir del control remoto es considerado un proceso IoT [Petrasch and Hentschke, 2016]. Por lo tanto, IoT permite a las cosas ser notificadas y controladas remotamente mediante una infraestructura de red existente, permitiendo una integración más directa entre el mundo físico y el mundo de los sistemas computacionales, lo cual resulta tanto en un incremento de eficiencia, precisión y mayor rentabilidad, como en la reducción de la intervención humana.

Es así que, para la integración de dispositivos IoT existen herramientas de desarrollo de flujo de código abierto como Node-RED, mismo que dispone de aplicaciones de programación de interfaz (APIs) y servicios en línea desarrollados por la tecnología emergente de IBM. Node-RED dispone de tres paneles, panel de nodos, panel de flujo e información, panel de ejecución y adicionalmente cuenta con una interfaz gráfica tipo dashboard [Ferencz and Domokos, 2020].

Los autores [Priyanka et al., 2020] integran un controlador LQR-PID para un sistema de vigilancia y control de flujo y presión en un sistema de transporte de flúidos, mencionan que el lazo cerrado de control local es monitoreado y controlado a través de un módulo IoT, mismo que es un puente entre los sensores embebidos y el centro de control. Asimismo, los autores [Lekić and Gardašević, 2018] muestran la implementación de una aplicación IoT usando un Raspberry Pi 3 modelo B. Para ello, primero, se obtienen las mediciones de los sensores DHT11 de temperatura y humedad y posteriormente se enlaza a la plataforma de IBM Bluemix. Finalmente, la aplicación en la nube muestra los datos recolectados desde el Raspberry Pi usando Node-RED y el protocolo WebSocket. Adicionalmente, los autores [Maheswari et al., 2020] implementan un PLC AC500 de la marca ABB con un Revelation-Pi que básicamente es un raspberry-pi industrial más robusto en hardware, para crear un módulo IoT capaz de otorgar eficiencia energética, latencia reducida, escalabilidad y mejor precisión, simplifica el monitoreo

por comunicación local y provee un procesamiento de datos obtenidos de diferentes sensores, proveyendo información crucial para el control futuro y la predicción de fallos.

Además, el protocolo MQTT (Message Queue Telemetry Transport), es uno de los más populares en comunicación IoT, puesto que es un protocolo ligero que puede ser ejecutado en dispositivos pequeños para mediciones y monitoreo, implementándolo a un sistema de medición de la calidad de energía de bajo costo. Aquí, un ESP8266 recolecta los datos de los sensores y los publica en un Broker MQTT usando Node-RED. Luego, los datos recibidos se los puede visualizar en una interfaz gráfica del usuario [Chanthakit and Rattanapoka, 2018]. La figura 2.3 muestra la arquitectura en la que se basa este sistema.

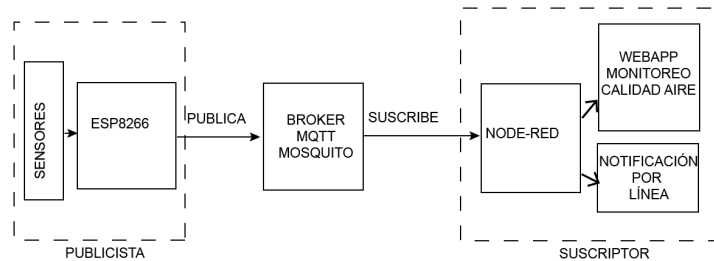


Figura 2.3: Arquitectura de un sistema IoT para el monitoreo de calidad del aire [Chanthakit and Rattanapoka, 2018]

Si se trabaja sobre una tarjeta raspberry-pi, se puede simular los datos de los sensores con un código python y con la ayuda del broker MQTT llamado Mosquito, los datos son transferidos en Node-RED y modificados de ser necesario, estos datos generados son almacenados en el cluster de una base de datos que luego pueden ser visualizados en una interfaz [Ferencz and Domokos, 2020].

En base a estos antecedentes, el presente proyecto busca contemplar los temas necesarios para el desarrollo de un sistema integral con un controlador óptimo LQR y un controlador óptimo LQI y así mantener una presión constante en la línea de descarga de un sistema de presión de agua siendo el algoritmo de control ejecutado en un Raspberry Pi que se enlaza a plataformas IoT para el monitoreo y control del proceso en tiempo real y se conecta con un PLC Siemens S7-1200 que se encarga de la lectura del transductor de presión y comanda al variador de frecuencia.

2.2. Formulación del Problema

La nueva tendencia tecnológica es la de implementar fábricas inteligentes con el desarrollo de la Industria 4.0, del internet de las cosas y de los sistemas embebidos, dando origen a satisfacer la necesidad de desarrollar equipos de monitoreo, control y adquisición de datos de manera remota que permitan enlazarse a dispositivos de control industrial como PLCs (controladores lógicos programables) para el control y automatización de procesos de una manera inteligente. Sin embargo, ¿se podría crear una solución integral accesible para muchas pequeñas y medianas fábricas que requieren procesos eficientes y rentables?

Consecuentemente, debido a la gran demanda de controladores eficientes en procesos industriales que ofrezcan un menor consumo y recursos computacionales en su implementación para obtener una respuesta con una acción de control óptima, el presente trabajo propone el diseño de un controlador óptimo para un proceso de control de presión que pueda ser implementado para el sector industrial usando equipos robustos, lo cual llevaría a conclusiones que servirían para contribuir y avanzar en el estudio de nuevas aplicaciones para controladores óptimos en lazos de control que usen motores para afectar una variable del proceso, puesto que actualmente en su mayoría, se utiliza el clásico controlador PID, mismo que en ciertos procesos genera fluctuaciones o sobre impulsos en la respuesta del sistema y conlleva tiempo de los técnicos en campo para su sintonización periódica.

Adicionalmente, este trabajo propone un diseño que a su vez es una solución innovadora, inteligente y rentable para el control y monitoreo remoto del sistema desde cualquier parte del mundo mediante el uso de sistemas embebidos que servirán como medio de comunicación con el internet y las herramientas IoT que permiten tener interfaces gráficas en computadores y dispositivos móviles y se propone además, el diseño de un sistema de control y monitoreo remoto usando herramientas IoT para mantener una presión constante en la línea de descarga de una bomba, logrando así que el sistema desarrollado sea escalable a otros procesos en la industria en general.

Capítulo 3

Metodología y Diseño

En este capítulo se desarrollará la metodología seleccionada, se realizará el diseño mecánico, diseño eléctrico, diseño del controlador, validación y simulación del sistema en su totalidad.

3.1. Metodología

El modelo en V se establece en el dominio de la ingeniería de sistemas y ha sido adaptado a las necesidades mecatrónicas y describe el procedimiento genérico para el diseño de sistemas mecatrónicos, mismos que deben ser especificados de acuerdo al diseño individual específico [Gausemeier and Moehringer, 2002]. Dicho modelo en V, modificado para el diseño mecatrónico, busca integrar varias etapas como el diseño, simulación e implementación física en el desarrollo de sistemas o productos mecatrónicos [Yu et al., 2013]. Es así que los sistemas mecatrónicos son productos multidisciplinarios, que incorporan una aplicación sinérgica e interactiva de varias ramas como mecánica, electrónica, control e ingeniería computacional [Mohebbi et al., 2019]. Así, el modelo en V base para el desarrollo de este proyecto se muestra en la figura 3.1.

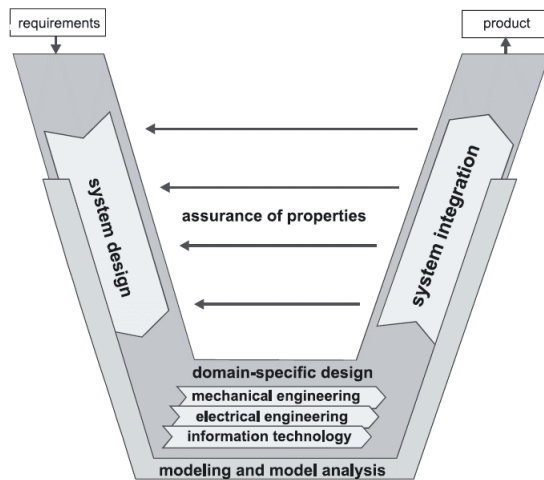


Figura 3.1: Modelo en V para el diseño mecatrónico [Gausemeier and Moehringer, 2002]

3.2. Requerimientos

El punto de inicio es una tarea de diseño individual, la cual se define y se describe con la ayuda de los requerimientos, que al mismo tiempo, representan las medidas de evaluación del producto final [Gausemeier and Moehringer, 2002].

Los requerimientos del sistema a desarrollarse son:

- a El sistema tendrá 1 bomba y un manifold de descarga donde se tengan los elementos sensores y elementos de visualización de campo.
- b El fluido a trabajar será agua a temperatura ambiente.
- c La bomba deberá cumplir los siguientes parámetros: altura manométrica total mínima de 40 metros, caudal mínimo de 1 m³/h, voltaje de entrada de 220V trifásico, el motor deberá ser cerrado y esta deberá ser centrífuga.
- d El voltaje de alimentación del sistema será 220V bifásico.
- e El sistema deberá ser capaz de estabilizar la presión en la línea en máximo 10 segundos.
- f El sistema hidráulico deberá tener las protecciones mínimas para garantizar su funcionamiento.
- g El sistema deberá tener un controlador óptimo que podrá estabilizar la presión en la línea de descarga en base al punto de consigna asignado, esta respuesta tendrá un error máximo de 1 Psi.
- h El tablero de control deberá ser compacto y contener todos los elementos de control, además, protecciones eléctricas necesarias para salvaguardar la integridad del sistema eléctrico-electrónico.
- i El algoritmo de control deberá ser ejecutado localmente en el sistema embebido que se enlazará al PLC.
- j Se deberá poder monitorear variables del proceso y controlar al sistema de manera remota.
- k La aplicación e interfaz gráfica para el monitoreo y control deberá ser compatible para PC y para dispositivo móvil como tableta o celular.

3.3. Diseño general del sistema

El objetivo del diseño del sistema es definir una solución integral que describa las características esenciales físicas y lógicas del producto futuro. Además, la función global de un sistema será dividida en otras sub funciones. Los principios adecuados de trabajo y/o los elementos de solución están asignados a estas sub funciones [Gausemeier and Moehring, 2002].

Básicamente, el lazo de control simplificado se lo presenta en la figura 3.2, en donde se lo muestra retro alimentado y se ingresa el punto de consigna de forma inalámbrica con la herramienta IoT. Este valor se compara con el valor del sensor y el error es ingresado al controlador y posteriormente este envía una señal de control al actuador del sistema el cual se compone del variador de frecuencia y la bomba y afecta directamente a la planta, es decir, al valor de presión en las líneas de descarga.

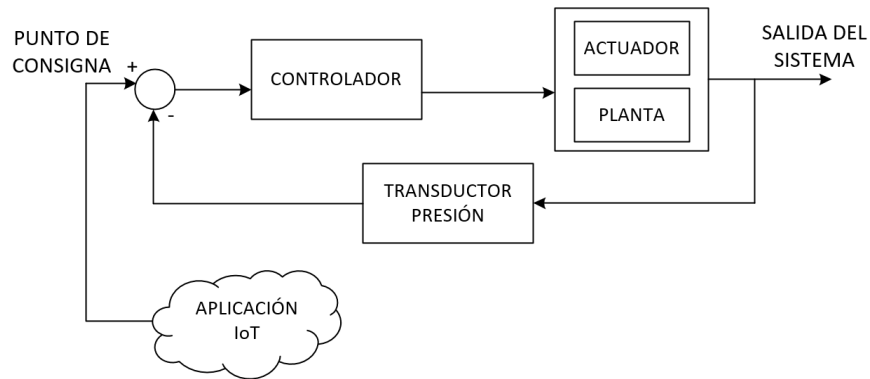


Figura 3.2: Diagrama de bloques general del sistema de presión constante

Una vez establecido el diagrama de bloques general del sistema, se propone la arquitectura integral y global del sistema. La figura 3.3 muestra la arquitectura de la solución integral planteada.

Además, se propone que el Raspberry Pi sea el encargado de recopilar y enviar los datos desde y hacia la aplicación IoT, y sea el dispositivo encargado de enlazar el sistema físico con el medio digital del internet. Así se podrá monitorear y controlar el sistema de forma inalámbrica desde un computador personal o desde cualquier dispositivo móvil con conexión a internet.

Por otra parte, la planta constará de un PLC que comanda al variador de frecuencia y este modificará la velocidad del motor de la bomba y así se presurizará la línea de descarga. Adicionalmente, el lazo de control se ejecutará en el Raspberry Pi, por lo que esta tarjeta deberá ser capaz de obtener el valor de retroalimentación del sensor y de enviar la señal de control al variador de frecuencia, de manera que se pueda aislar la parte de control y de fuerza y con el fin de añadir robustez al sistema, se usará un PLC como medio físico de adquisición de datos del sensor de la planta y lectura de los elementos de mando y control. Asimismo, se usará el actuador de forma

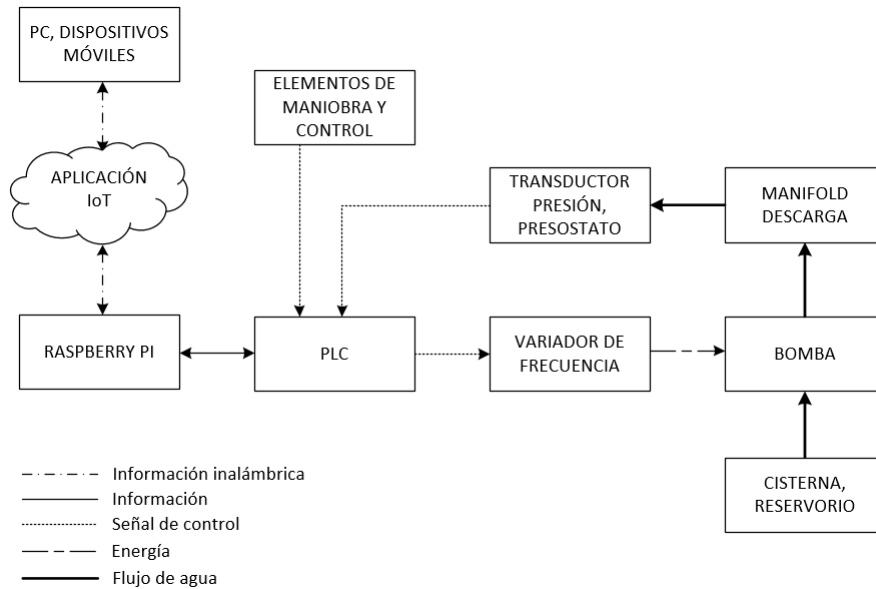


Figura 3.3: Arquitectura integral del sistema de presión constante a desarrollarse

alámbrica para el envío de señales de control. Por consiguiente, el sensor y el variador de frecuencia irán conectados al PLC y el Raspberry Pi se enlazará al PLC de forma alámbrica, leyendo y escribiendo los registros internos del PLC mediante protocolos de comunicación.

Por otra parte, el manifold de descarga de la bomba será construido con tubería que soporte la presión adecuada y tendrá el espacio y tomas necesarias para colocar el transductor de presión o elemento sensor. El sistema hidráulico constará de los elementos necesarios mínimos de seguridad y funcionamiento.

En conclusión, el concepto de solución que ha sido propuesto y que engloba varias áreas de la ingeniería será resuelto separadamente de manera más detallada conforme al dominio específico y se elaborará su diseño y cálculos necesarios para garantizar un rendimiento funcional [Gausemeier and Moehring, 2002].

3.4. Diseño mecánico

El diseño mecánico del sistema abarcará la selección de los elementos necesarios para la construcción de la planta en base a criterios técnicos y de rentabilidad e implicará la selección de la bomba, el sistema de tuberías y selección de válvulas. Así, el desarrollo del diseño mecánico y la construcción del sistema se basarán en el concepto de que este es un sistema didáctico para pruebas y más no se lo implementará en una edificación real, por lo que la longitud de tuberías será menor a 2 metros.

3.4.1. Selección de bomba

Las bombas con velocidad variable producen un número infinito de características de la bomba, las mismas que pueden hacer que esta opere virtualmente en cualquier punto de la curva. Además, cuando una bomba se conecta a un sistema de tuberías, ella trabajará en el punto de equilibrio entre la bomba y el sistema de tuberías. Es así que, en este punto de equilibrio, la energía suministrada por la bomba es igual a las pérdidas debido a la resistencia en el sistema. Cabe señalar que las características de la bomba son representadas por la curva de altura manométrica versus caudal [Nesbitt, 2006].

Uno de los requisitos es que la altura dinámica total mínima sea de 40 metros, razón por la cual, aproximadamente la presión en la línea de descarga está comprendida entre 0 y 70 Psi, con un flujo mínimo de 1 m³/h. Además, el fluido con el cual se va a trabajar es agua a temperatura ambiente y se deberá tomar en cuenta que la bomba debe ser trifásica para poder ser controlada con variador de frecuencia y el voltaje del sistema será 220 voltios bifásico. En este caso, las bombas propuestas son centrífugas y periféricas.

- La primera opción es una bomba de la marca Thebe modelo TH16AL con un motor de 1HP, 220V, 3F. La figura 3.4 muestra la curva TDH versus caudal de la bomba correspondiente y se ha enfatizado el punto de mayor eficiencia de la bomba.
- La segunda opción es una bomba Ebara PRA 106T con un motor de 1HP, 220V, 3F. La figura 3.5 muestra la curva TDH versus caudal de la bomba en cuestión y se ha enfatizado el punto de mayor eficiencia de la bomba.

Además, se asume que las pérdidas por fricción son cero, puesto que al ser un sistema didáctico pequeño, la longitud de tubería no excederá los 2 metros

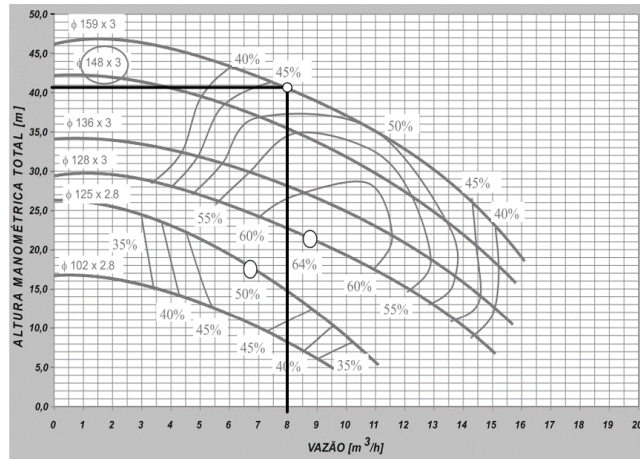


Figura 3.4: Curva característica de la bomba Thebe

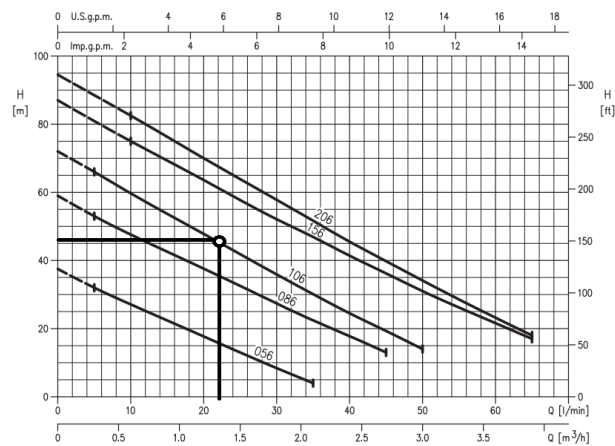


Figura 3.5: Curva característica de la bomba Ebara PARA 106T

y a su vez la cantidad de codos y uniones es mínima. Por ende, en el caso de realizar el cálculo de pérdidas por fricción, el valor sería muy pequeño, de manera que se lo puede omitir.

Es importante considerar también que el criterio principal para la selección de la bomba adecuada, fue el del costo-beneficio, ya que ambas bombas propuestas tienen similares características. Sin embargo, la bomba marca Ebara puede alcanzar una altura mayor con una mayor eficiencia, cumple

con los parámetros de caudal solicitados en el requerimiento “c”, e inclusive es más económica que la bomba Pedrollo. Por el otro lado, la bomba Thebe puede alcanzar una altura dinámica total de 40 metros. Sin embargo, está fuera de su punto de eficiencia. Por ende, la bomba Ebara PRA 106T fue la seleccionada.

3.4.2. Diseño de tubería y montaje del sistema hidráulico

El diseño del sistema hidráulico completo del sistema se lo hizo en SOLIDWORKS, de manera que se obtiene el modelo que se indica en la figura 3.6. Aquí, se puede visualizar que se tiene una estructura metálica que soporta la bomba y el reservorio de agua, el mismo que es de plástico translúcido. La tubería es PVC pegable para presión de 1/2 pulgada para descarga y 1” para la succión en marca Rival. Esta tubería soporta hasta 320 Psi en 1” y 180 Psi en 1/2”. A la descarga de la bomba se tiene una válvula a la cual se puede dar apertura manualmente, disminuyendo así la presión en la línea de descarga y generando perturbaciones para el controlador. Esto se vuelve a recircular, de manera que no se tiene que volver a llenar el reservorio.

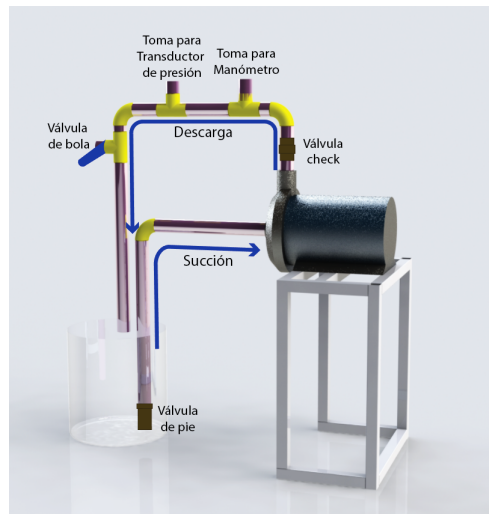


Figura 3.6: Sistema hidráulico modelado en 3D

Considerando que la succión de la bomba es negativa, se incluye una válvula de pie de bronce del diámetro correspondiente en la línea de succión. Además, el manifold de la bomba tiene las tomas necesarias para válvulas: 1 transductor de presión y 1 manómetro.

Se puede apreciar el diagrama P&ID (Piping and Instrumentation Diagram por sus siglas en inglés) en la figura 3.7.

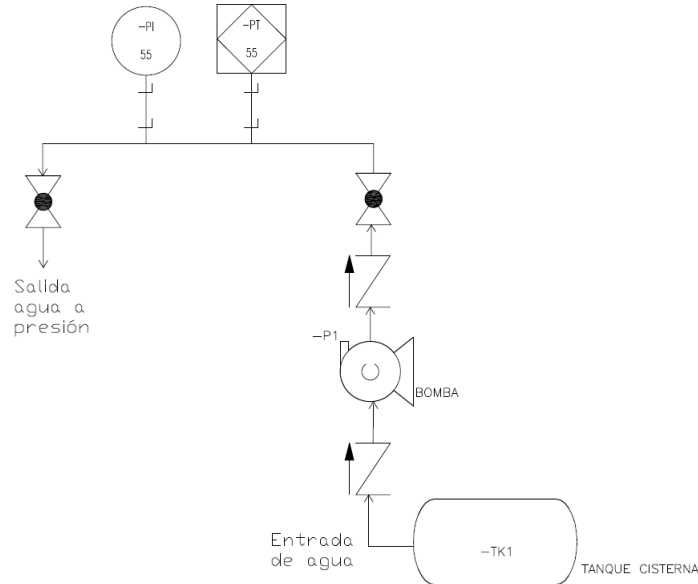


Figura 3.7: Diagrama P&ID del sistema

3.5. Diseño eléctrico

El sistema tendrá un tablero de control donde se montarán todos los elementos de protección, mando, control y fuerza, para lo cual se procede a realizar la selección de equipos para el control del sistema que requiere un variador de frecuencia para comandar la velocidad a la que girará el motor de la bomba; elementos de protección eléctrica; fuentes de voltaje para energizar los equipos y elementos de mando correspondientes; un transductor de presión para obtener la presión de la línea de descarga; un controlador lógico programable con sus módulos de expansión analógicos y la tarjeta Raspberry.

3.5.1. Selección del variador de frecuencia

Se procede a la selección del variador de frecuencia con los siguientes requisitos mínimos de operación:

- Entrada bifásica 220 Voltios
- Corriente de salida 4.2 Amperios
- Entradas digitales de control
- Entrada analógica
- Tamaño compacto
- Control de velocidad/frecuencia
- Protección IP 20
- Frecuencia máxima 200 Hz.

Las características de los equipos propuestos se presentan en las tablas 3.1 y 3.2. Ambos equipos tienen características similares. Sin embargo, el seleccionado es el de la marca Allen Bradley, puesto que este dispone de dos entradas analógicas e incorpora un ventilador para la refrigeración del equipo, lo cual impide que los demás elementos dentro del tablero se calienten. Además, es de una marca fiable ya que su robustez y fiabilidad definieron su selección para esta aplicación. La figura 3.8 muestra el aspecto físico del variador de frecuencia seleccionado.

Tabla 3.1: Características técnicas del variador de frecuencia WEG CFW100040SDPLZ.

Característica	Valor
Voltaje nominal de entrada	200-240 Voltios AC, 1 Fase
Potencia de motor a conectar	1HP
Corriente de salida nominal	4.0 Amperios
Frecuencia máxima	600 Hz
Protección	IP20
Entradas digitales programables	4 entradas digitales
Entradas analógicas	1 Entrada analógica 0-10VDC
Tamaño	96x132x122 mm (ancho,altura,profundidad)

3.5.2. Selección de la fuente de poder

Se procede a la selección de la fuente de poder que proveerá de 24 voltios en corriente directa para el transductor de presión, elementos de maniobra

Tabla 3.2: Características técnicas del variador de frecuencia Allen Bradley 22A-A4P5N104.

Característica	Valor
Voltaje nominal de entrada	200-240 Voltios AC, 1 Fase
Potencia de motor a conectar	1HP
Corriente de salida nominal	4.5 Amperios
Frecuencia máxima	240 Hz
Protección	IP20
Entradas digitales programables	4 entradas digitales
Entradas analógicas	2 Entradas analógicas voltaje-corriente
Tamaño	80x152x136 mm (ancho,altura,profundidad)



Figura 3.8: Variador de frecuencia Allen Bradley 22A-A4P5N104

y control. La selección de la fuente se la hizo en base a la corriente necesaria requerida y que sea para montaje en riel din.

La fuente seleccionada es una fuente de la marca Delta DRP012V030W1AZ de 30 Watts mostrada en la figura 3.9. El voltaje de alimentación monofásico es de 85-264VAC, con un voltaje de salida de 24VDC, obteniendo una corriente máxima de 2.5 Amperios.

3.5.3. Selección del controlador lógico programable

Se requiere de un controlador lógico programable o PLC capaz de ser el medio físico para la obtención y envío de señales de control conforme al algoritmo de control establecido. Además, en base a las necesidades del sistema, el PLC deberá constar de mínimo 2 entradas digitales a 24VDC



Figura 3.9: Fuente de poder Delta DRP012V030W1AZ

configurables; 1 entradas analógica de voltaje; 2 salidas tipo relé; 1 salida analógica de voltaje y 1 puerto ethernet para comunicación. Adicionalmente, se recomienda que el PLC disponga de fuente interna.

El PLC propuesto es un Siemens S7-1200 1212C AC-DC-RLY, mismo que cumple con los requerimientos. La tabla 3.3 muestra las características del mismo. Cabe recalcar que se usará un módulo de expansión tipo placa de señales para la salida analógica de voltaje, el mismo que irá montado en la parte frontal del PLC. La figura 3.10 muestra el aspecto físico del PLC seleccionado.

Tabla 3.3: Características técnicas del PLC Siemens S7-1200 1212C AC-DC-RLY.

Característica	Valor
Voltaje de alimentación	85-264 Voltios AC, 1 Fase
Consumo de corriente	40mA a 240VAC
Reloj en tiempo real	RTC 480 horas de respaldo
Entradas digitales	8 integradas, (4 rápidas) source o sink
Salidas digitales	6 tipo relé, hasta 2A c/u
Entradas analógicas	2 entradas analógicas 0-10VDC
Interfaz	Profinet
Tipo de comunicación	Comunicación S7
Protecciones eléctricas	EMC
Protección física	IP 20
Medidas	90x100x75mm



Figura 3.10: PLC Siemens S7-1200 1212C AC-DC-RLY

3.5.4. Selección del transductor de presión

Al disponer de entradas analógicas de voltaje en el PLC, se requiere un transductor de presión con salida de voltaje entre 0 y 10 VDC. Adicionalmente, en el sistema, el rango de operación de presión está entre 0 y 70 PSI aproximadamente. Por otra parte, la toma de conexión es de $\frac{1}{4}$ " NPT. Se trabajará con agua a temperatura ambiente y el voltaje de alimentación del transductor deberá ser de 24VDC.

Se selecciona un transductor de presión de la marca Pixsys Electronics con las características presentadas en la tabla 3.4. La figura 3.11 muestra el aspecto físico del sensor seleccionado que cumple con los requerimientos del caso.

Tabla 3.4: Características técnicas del sensor Pixsys Electronics ECT-NAT.

Característica	Valor
Voltaje de alimentación	9-30VDC
Precisión	0.5 % FS
Salida de voltaje	0-10VDC
Protección	IP65
Conexión a proceso	$\frac{1}{4}$ " NPT
Material	Acero inoxidable AISI 316L
Rango de operación	0-145PSI



Figura 3.11: Sensor de presión Pixsys Electronics ECT-NAT.

3.5.5. Selección de la tarjeta Raspberry Pi

Se selecciona la tarjeta Raspberry Pi 3 B+ ya que presenta mejores capacidades de procesamiento, Wifi de banda dual y puerto ethernet tipo Gigabit. La tabla 3.5 muestra las características en detalle y la figura 3.12 su aspecto físico. Esta tarjeta será la encargada de enlazarse con la aplicación IoT y con el PLC, de manera que deberá tener la posibilidad de conectarse a una red WiFi y simultáneamente tener una dirección IP fija con la que se enlazará al PLC.

Tabla 3.5: Características técnicas de la tarjeta Raspberry Pi 3 B+.

Característica	Valor
Tipo de procesador	BCM2837B0, Cortex-A53 (ARMv8) 64-bit, SoC @ 1.4GHz , 1GB LPDDR2 SDRAM
Conectividad WiFi	Wifi dual 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac
Puerto Ethernet	Ethernet Gigabit sobre USB 2.0 hasta 300 Mbps
Voltaje de alimentación	5V
Corriente de consumo	2.5Amperios

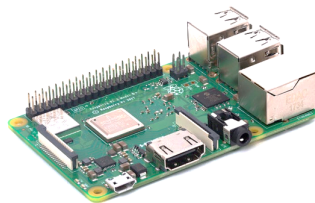


Figura 3.12: Raspberry Pi 3 B+.

3.5.6. Diseño de planos eléctricos

Para empezar el diseño de los planos eléctricos, se debe partir en base a los diagramas de conexión que establecen los fabricantes, en este caso los diagramas a considerar son el del variador de frecuencia presentado en la figura 3.13 y el del PLC presentado en la figura 3.14, los demás elementos solamente requieren una polarización adecuada.

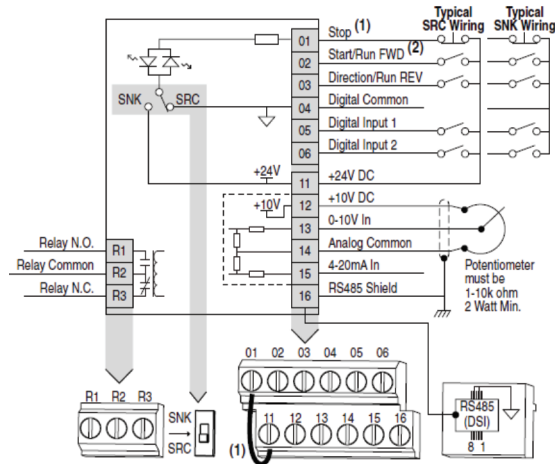


Figura 3.13: Diagrama de conexiones del variador de frecuencia Allen Bradley 22A-A4P5N104.

Por otra parte, el variador de frecuencia dispone de entradas digitales que serán activadas desde el PLC usando relés de interfaz. Se usarán las entradas 02 para arrancar el variador de frecuencia en base a la entrada analógica externa y 05 para arrancar el variador de frecuencia en operación JOG. Es decir, con una frecuencia de trabajo establecida, esto se usará para el caso de operación manual. Además, la entrada analógica del variador de frecuencia será usada para variar la frecuencia de trabajo de acuerdo a la salida entregada por el controlador. Este valor analógico se comprende entre 0 y 10 voltios. La referencia de voltaje será el puerto 14 en el variador y la entrada de voltaje irá conectada al puerto 13.

Una vez que se tienen los diagramas de conexión de los elementos de acuerdo a cómo lo estipula el fabricante, se procede con el diseño de los planos eléctricos, mismos que fueron desarrollados en AutoCAD Electrical 2019, en base a criterios técnicos y normas de diseño, los planos eléctricos completos se los puede encontrar en el Anexo 1. La figura 3.15 muestra el plano de fuerza

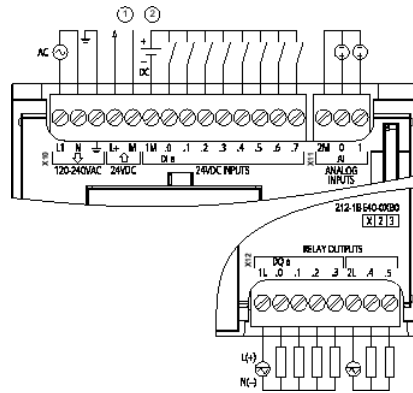


Figura 3.14: Diagrama de conexión del PLC Siemens S7-1200 1212C AC-DC-RLY.

simplificado sin el cajetín correspondiente, mismo que contempla el esquema eléctrico del variador de frecuencia y las protecciones en las líneas de voltaje. La figura 3.16 muestra el plano de control, el cual abarca los diagramas de conexión para el control del PLC, entradas analógicas y digitales del variador de frecuencia, fuente de voltaje, transductor, elementos de maniobra y protecciones de control.

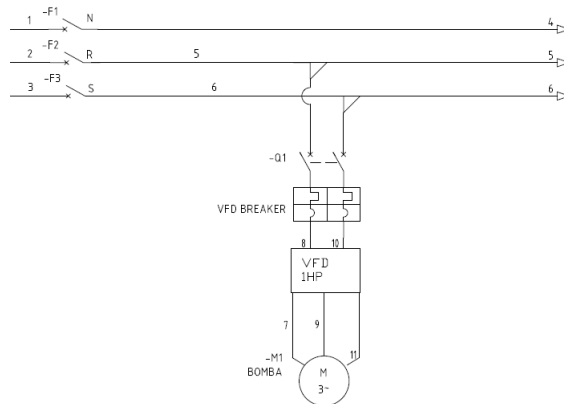


Figura 3.15: Plano eléctrico de fuerza del tablero de control.

Para la parte de fuerza, se usará un disyuntor termo magnético de 2 polos para las fases, este será de 25 Amperios con capacidad de ruptura en AC de

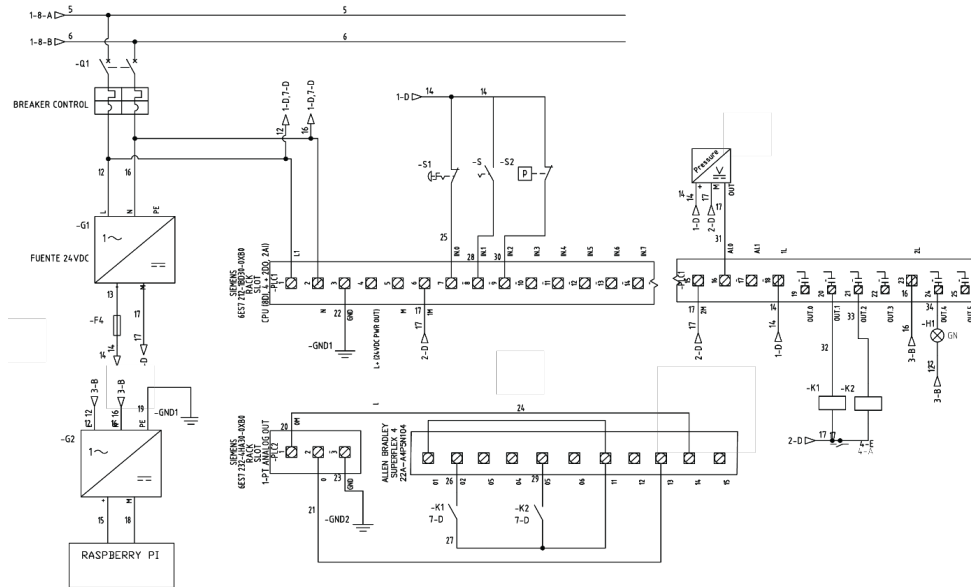


Figura 3.16: Plano eléctrico de control del tablero de control.

10 KA. Posteriormente, se usará un disyuntor bipolar de 20 Amperios, este dimensionamiento se lo hizo conforme lo estipula el fabricante, la corriente del breaker de protección debe ser entre 2 y 4 veces la corriente nominal a la entrada. La corriente de entrada es de 9.7 Amperios.

La parte de control dispone de un breaker principal de protección de 6 Amperios. Aquí se contemplan todos los elementos de control del sistema, como PLC, fuentes de poder, Raspberry Pi, las entradas digitales y analógicas del variador de frecuencia entre otros elementos de maniobra y control. Este plano eléctrico usa codificaciones de líneas de cables. Por ejemplo, la tierra física lleva el nombre de “GND”. 14 y 17 son los números de cable para la salida de la fuente de voltaje, siendo +24VDC y -24VDC respectivamente.

Se usarán 2 entradas del PLC: una destinada al paro de emergencia, primordial en cualquier sistema de control, y un selector de 2 posiciones para encendido y apagado. A la salida del PLC se tendrá una luz piloto que indicará en el tablero el estado del sistema, y dos bobinas de relé destinadas al control del variador de frecuencia. El módulo de expansión de salida analógica del PLC enviará el voltaje a la entrada analógica del variador de frecuencia en base a la salida del controlador.

3.6. Diseño del sistema IoT

El sistema tendrá una plataforma IoT desde la cual se podrá controlar remotamente el sistema, encenderlo, establecer modos de operación y visualizar variables de proceso.

3.6.1. Protocolo MQTT

Muchas plataformas IoT en el mercado son pagadas, debiendo cancelar una suscripción mensual o anual para proceder acceder a sus servicios. Por esta razón, se optó por obtener un servidor MQTT de uso propio, donde no se dispongan limitaciones de capacidad de publicación y suscripción de datos y que permita el enlace entre dispositivos.

MQTT (Message Queue Telemetry Transport por sus siglas en inglés), fue diseñado por IBM en 1999 para una comunicación M2M (Machine to machine por sus siglas en inglés) con el objetivo de proveer un protocolo de mensajes de publicación-suscripción con un ancho de banda, tamaño de codificación, y consumo de potencia lo más pequeños posibles. El protocolo MQTT dispone de tópicos, los cuales son nombres jerárquicos separados con un slash “/” [Babovic et al., 2016].

Este, al ser tan liviano, puede ser soportado por los dispositivos más pequeños para medición y monitoreo e inclusive puede enviar datos en redes intermitentes. Adicionalmente, es de software abierto, lo que lo hace fácil de adaptarlo a una variedad de necesidades de mensajes y de comunicación. Este protocolo, en comparación con HTTP, está orientado a los datos y transfiere contenido de datos como un vector de bytes. Además, MQTT utiliza un patrón de mensajes publicación/suscripción que tiene un acoplamiento suelto, que hace que los clientes no necesiten saber sobre la existencia de otros dispositivos, sino que solamente se preocupan por el contenido a ser entregado o receptado [Lampkin et al., 2012].

Un cliente MQTT recolecta información de un dispositivo de telemetría, se conecta a un servidor de mensajería y usa tópicos tipo string para publicar la información de manera que permite a otros clientes o aplicaciones obtener dicha información y también puede suscribirse a un tópico. Por otro lado, el bróker MQTT es un servidor que implementa el protocolo MQTT y es un mediador de comunicación entre los clientes [Lampkin et al., 2012].

En el caso de la aplicación a desarrollarse, y con el fin de poder publicar y suscribirse, se requiere de un servidor MQTT, mismo que se lo configurará en un computador virtual provisto por Google Cloud. De modo que, sin importar en qué red o ubicación estén los dispositivos o clientes, estos puedan

suscribirse y publicar a un solo servidor MQTT, generando la capacidad de conectividad remota. Las figuras 3.17 y 3.18 muestran la arquitectura resumida que se implementará en este sistema.

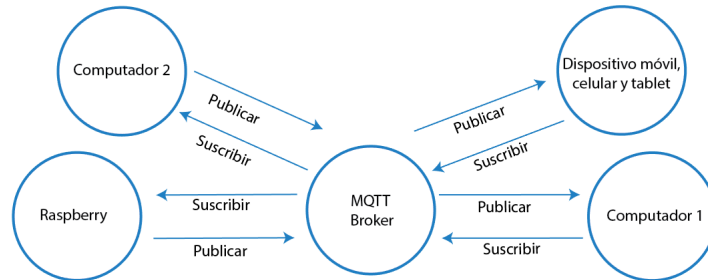


Figura 3.17: Arquitectura de funcionamiento de MQTT.

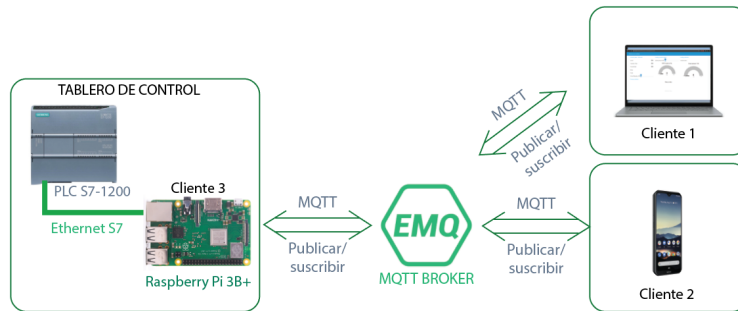


Figura 3.18: Esquema para suscribir y publicar a broker MQTT del sistema.

3.6.2. Implementación de broker MQTT

Con el fin de poder disponer de un broker MQTT se requiere que este esté corriendo sobre un computador o servidor que brinde disponibilidad y redundancia, por esta razón se implementó un computador virtual en la plataforma de Google Cloud. Google Cloud Platform es una herramienta tipo suite de servicios de computación en la nube. Corre sobre la misma infraestructura que Google usa para sus productos como Google Search, Gmail y Youtube. La figura 3.19 muestra la interfaz de usuario de esta plataforma y la figura 3.20 muestra la instancia de máquina virtual creada en la plataforma de Google Cloud.

Dentro de la suite de Google Cloud, se puede encontrar la herramienta Compute Engine, que básicamente permite tener máquinas virtuales corriendo sobre los servidores de Google. En este caso se creó una máquina virtual con las siguientes características:

- Nombre de instancia: atsimec
- Tipo de máquina: g1-small (1 vCPU, 1,7 GB de memoria)
- Plataforma de CPU: Intel Broadwell
- Imagen de arranque: Ubuntu-1804-bionic-v20200430 – 10GB
- Sistema operativo: UBUNTU 18.04.4 LTS

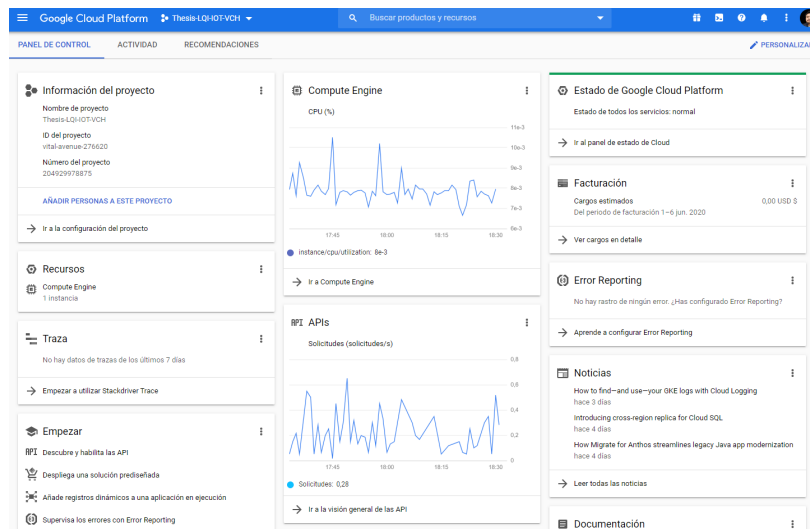


Figura 3.19: Google Cloud Platform.

Instancias de VM

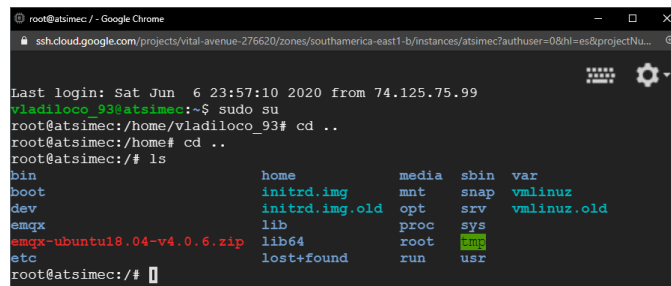
CREAR INSTANCIA IMPORTAR VM ACTUALIZAR INICIAR DETENER RESTABLECER

Filtrar las instancias de VM

Nombre	Zona	Recomendación	Usada por	IP interna	IP externa	Conectar
atsimec	southamerica-east1-b			10.158.0.4 (nic0)	34.95.185.169 L2	SSH

Figura 3.20: Instancia creada en Google Cloud Platform.

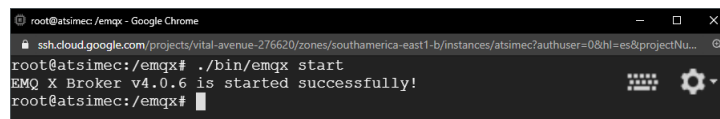
Esta instancia de máquina virtual se la creó de manera que sobre ésta, se pueda instalar el broker MQTT para poder enlazar los dispositivos requeridos, y a su vez, sobre esta máquina virtual se ha instalado EMQX, el cual es un broker open source de MQTT. En Ubuntu, ingresamos el comando `ls`, para poder visualizar la lista del contenido de directorios, como se muestra en la figura 3.21 de manera que se puede constatar que EMQX ha sido instalado correctamente en la máquina virtual.



```
root@atsimec / - Google Chrome
ssh.cloud.google.com/projects/vital-avenue-276620/zones/southamerica-east1-b/instances/atsimec?authuser=0&hl=es&projectNu...

Last login: Sat Jun  6 23:57:10 2020 from 74.125.75.99
vladiloco_93@atsimec:~$ sudo su
root@atsimec:/home/vladiloco_93# cd ..
root@atsimec:/home# cd ..
root@atsimec:/# ls
bin          home          media         sbin         var
boot         initrd.img    mnt           snap         vmlinuz
dev          initrd.img.old  opt          srv          vmlinuz.old
emqx         lib           proc          sys
emqx-ubuntu18.04-v4.0.6.zip  lib64        root         snap
etc          lost+found    run          usr
root@atsimec:/#
```

Figura 3.21: Lista de contenido de directorios en máquina virtual con Ubuntu.



```
root@atsimec/emqx - Google Chrome
ssh.cloud.google.com/projects/vital-avenue-276620/zones/southamerica-east1-b/instances/atsimec?authuser=0&hl=es&projectNu...

root@atsimec:/emqx# ./bin/emqx start
EMQ X Broker v4.0.6 is started successfully!
root@atsimec:/emqx#
```

Figura 3.22: Arranque de broker MQTT en Ubuntu.

Una vez instalado ya el broker MQTT, debemos arrancarlo para que se empiece a ejecutar y poder acceder al mismo de forma remota, igualmente. Cabe recalcar que, se debe usar un firewall externo en la máquina virtual para poder acceder al broker MQTT, se debe abrir el puerto de igual manera en Ubuntu, siendo el puerto en cuestión, el 18083. Ya que el broker MQTT esté arrancado, se puede acceder a su dashboard de forma remota, usando la IP externa de la máquina virtual anteriormente creada y accediendo al puerto 18083.

La figura 3.23 muestra el dashboard del broker MQTT. Aquí se puede monitorear en tiempo real algunas métricas de clientes conectados y número de tópicos, entre otros. La figura 3.24 muestra el estado de los clientes actuales. Es el caso de la Raspberry y del computador donde se tiene

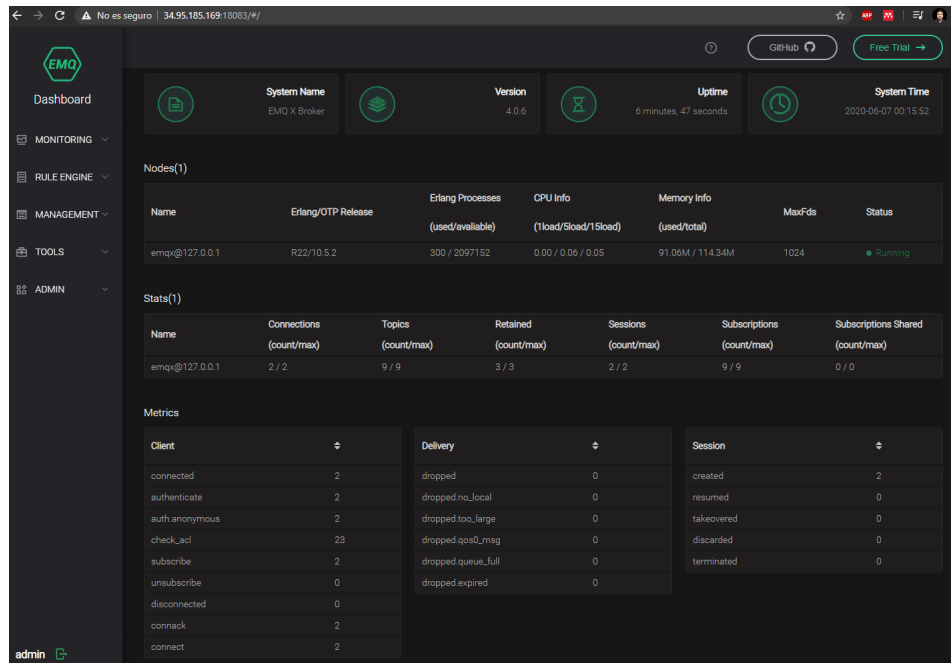


Figura 3.23: Dashboard de EMQX.

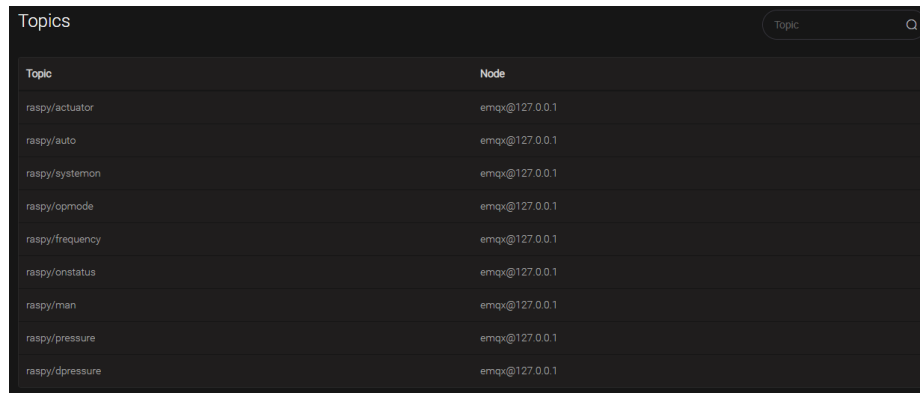
instalado Node-RED y se usará el dashboard del mismo. Adicionalmente, la figura 3.25 muestra los tópicos a tratarse de acuerdo a la aplicación en cuestión. Dichos tópicos podrán ser publicados o suscritos por el Raspberry o por cualquier dispositivo que desee acceder o modificar el mismo.

		emqx@127.0.0.1		Client ID				
Client ID	Username	IP Address	Keepalive(s)	Expiry Interval(s)	Subscriptions Count	Connect Status	Created At	Operation
rasp-client	undefined	181.199.47.99.62...	60	0	5	CONNECTED	2020-06-07 00:09:11	Kick Out
test-client	undefined	181.199.47.99.43...	60	0	4	CONNECTED	2020-06-07 00:09:08	Kick Out

Figura 3.24: Clientes en el broker MQTT.

3.6.3. Selección de plataforma IoT

Se seleccionó una plataforma IoT que permite tener una interfaz gráfica compacta y personalizada para el uso en dispositivos móviles. Esta es



The screenshot shows a web interface titled 'Topics' with a search bar. Below the search bar is a table with two columns: 'Topic' and 'Node'. The table lists several topics, all of which are associated with the same node: 'emqx@127.0.0.1'.

Topic	Node
raspy/actuator	emqx@127.0.0.1
raspy/auto	emqx@127.0.0.1
raspy/systemon	emqx@127.0.0.1
raspy/opmode	emqx@127.0.0.1
raspy/frequency	emqx@127.0.0.1
raspy/onstatus	emqx@127.0.0.1
raspy/man	emqx@127.0.0.1
raspy/pressure	emqx@127.0.0.1
raspy/dpressure	emqx@127.0.0.1

Figura 3.25: Tópicos en el broker MQTT.

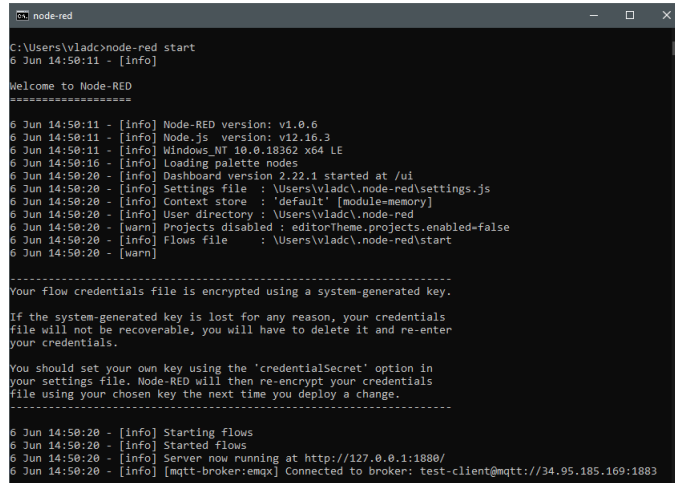
compatible para Android y iOS. Tal es el caso de Blynk. Blynk, la cual es una plataforma de IoT independiente del hardware con aplicaciones móviles de marca blanca, nubes privadas, administración de dispositivos, análisis de datos y aprendizaje automático que puede ser descargada desde la tienda de aplicaciones de cada sistema operativo en dispositivos móviles. Dispone de una variedad de widgets pre diseñados para construir aplicaciones iOS y Android y así controlar dispositivos electrónicos, monitorear datos de sensores, recibir notificaciones y muchas cosas más.

Para la interfaz de control IoT desde un computador, se seleccionó el dashboard de Node-RED. Este es un módulo que provee Node-RED para crear un tablero de instrumentos en tiempo real y así poder visualizar los datos obtenidos o enviados desde Node-RED. Node-RED es una herramienta de desarrollo de código abierto basada en flujos, que permite la integración de dispositivos IoT [Lekić and Gardašević, 2018]. Este dashboard se ejecuta sobre un servidor local, una vez que es instalado en el computador, mismo que se destina para el monitoreo y control del sistema.

3.6.4. Diseño de la interfaz en el dashboard de Node-RED

El diseño de la interfaz parte de la creación de un tablero de instrumentos o dashboard. Estos nodos son configurados desde Node-RED. Posteriormente, se establecen los tamaños de cada elemento a utilizar, se accede al tablero desde <https://127.0.0.1:1880/ui> o desde el servidor local que se determina cuando se inicia Node-RED en el computador y desde ahí se podrá modificar el panel de instrumentos o a su vez monitorear y controlar el proceso. La

figura 3.26 muestra la ventana de comandos de Windows, de donde se arrancó Node-RED.



```
C:\Users\vlad>node-red start
6 Jun 14:50:11 - [Info]
Welcome to Node-RED
=====
6 Jun 14:50:11 - [Info] Node-RED version: v1.0.6
6 Jun 14:50:11 - [Info] Node.js version: v12.16.3
6 Jun 14:50:11 - [Info] Windows_NT 10.0.18362 x64 LE
6 Jun 14:50:16 - [Info] Loading palette nodes
6 Jun 14:50:20 - [Info] Dashboard version 2.22.1 started at /ui
6 Jun 14:50:20 - [Info] Settings file : \Users\vlad\.node-red\settings.js
6 Jun 14:50:20 - [Info] Context store : 'default' [module=memory]
6 Jun 14:50:20 - [Info] User directory : \Users\vlad\.node-red
6 Jun 14:50:20 - [warn] Projects disabled : editorTheme.projects.enabled=false
6 Jun 14:50:20 - [Info] Flows file : \Users\vlad\.node-red\start
6 Jun 14:50:20 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
6 Jun 14:50:20 - [Info] Starting flows
6 Jun 14:50:20 - [Info] Started flows
6 Jun 14:50:20 - [Info] Server now running at http://127.0.0.1:1880/
6 Jun 14:50:20 - [Info] [mqtt-broker:emqx] Connected to broker: test-client@mqtt://34.95.185.169:1883
```

Figura 3.26: Inicio de Node-RED en computador con Windows.

La interfaz mostrada en la figura 3.27 consta de 3 pulsadores tipo interruptor para encendido del sistema, selección de modo automático y modo manual. Se dispone igualmente de una casilla de texto para indicar el estado del sistema y el modo de operación seleccionado. Adicionalmente, los valores analógicos de presión son visualizados en un indicador: uno tipo gauge. El histograma muestra los datos de presión durante los últimos 30 minutos de operación y se tiene un ingreso numérico tipo slider para el punto de consigna del controlador o la presión de trabajo requerida. En caso de arrancar el sistema en modo manual, se dispone de igual manera de un slider para ingresar la frecuencia deseada en manual del variador de frecuencia.

La figura 3.28 muestra el envío y recepción de datos, los cuales posteriormente son enviados a los widgets configurados del tablero de Node-RED.

3.6.5. Diseño de interfaz en Blynk

La interfaz en Blynk de igual manera se basa en widgets ya creados que son asignados a variables virtuales, las mismas que luego serán accedidas desde el servidor MQTT de la aplicación con librerías. La figura 3.29 muestra

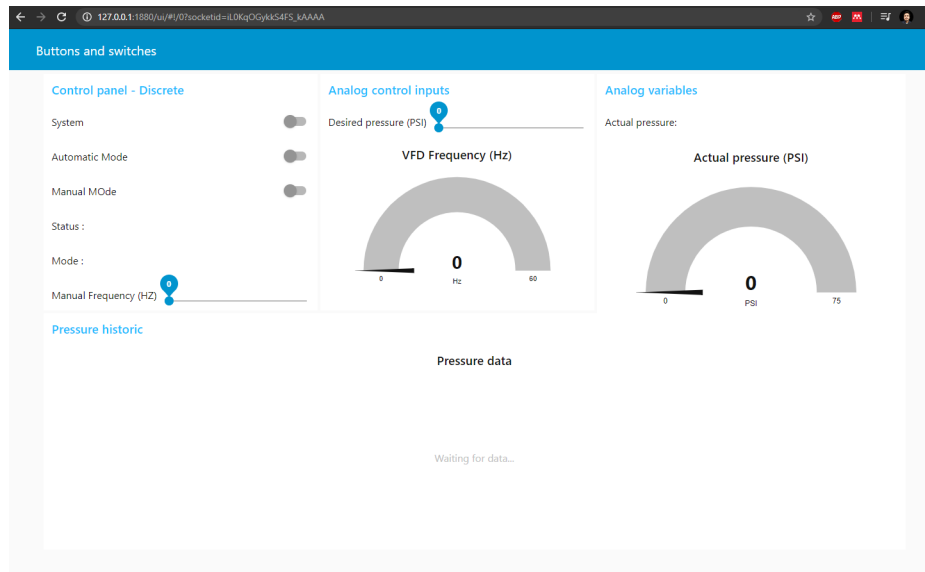


Figura 3.27: Diseño de interfaz en el tablero de Node-RED.

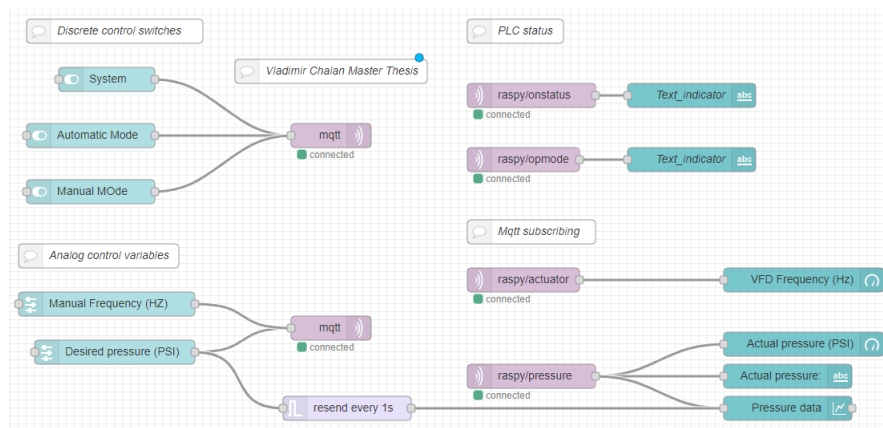


Figura 3.28: Configuración de dashboard en Node-RED.

la interfaz desarrollada en Blynk, con la misma cantidad de elementos y funcionalidad que la interfaz en Node-RED, de manera que el usuario de la plataforma no tenga inconvenientes con su operación. De hecho, esta plataforma IoT es netamente para teléfonos inteligentes por lo cual su

interacción con el mismo es sencilla y ofrece una rápida configuración.

Así, una vez desarrolladas las interfaces en las diferentes plataformas IoT, se deben enlazar las variables de las mismas con la tarjeta Raspberry Pi, para que esta sea la que gestione los datos de envío y recepción hacia el controlador y hacia la planta.

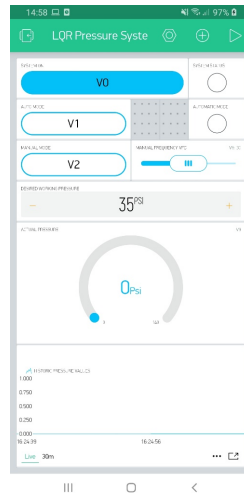


Figura 3.29: Diseño de interfaz en plataforma Blynk.

3.6.6. Configuración de Node-RED en Raspberry

Como se mencionó anteriormente, en el capítulo de introducción, Node-RED es una herramienta de programación para enlazar hardware con dispositivos, APIs y servicios en línea de una manera nueva e interesante. Es así que Node-RED está construido sobre Node.js, aprovechando su modelo sin bloqueo controlado por eventos y puede ser ejecutado localmente, en un dispositivo como Raspberry Pi o en la nube como en Amazon Web Services.

Sin embargo, antes de instalar y arrancar Node-RED en Raspberry Pi, se deberá asegurar que la Raspberry Pi tenga instalado Noobs, el cual es un sistema operativo fácil de manejar que contiene Raspbian y LibreELEC y posteriormente se deberá proceder a configurar los parámetros de red.

Cabe señalar que Raspberry Pi usa el registro “dhcpcd” para configurar TCP/IP alrededor de todas las interfaces de red. El sistema operativo Raspbian configura por defecto las interfaces de red por DHCP, lo que provoca una dirección IP privada automática. En este caso, se requerirá

que la dirección IP de la interfaz Ethernet sea fija para el enlace con el PLC y de igual manera se debe priorizar la interfaz Wifi sobre la interfaz Ethernet para la conexión a internet. Estos parámetros son modificados en el fichero “dhcpcd.conf” como se muestra en la figura 3.30. La dirección IP de la interfaz Ethernet de la Raspberry Pi será 192.168.10.20.

```
pi@raspberrypi: ~  
File Edit Tabs Help  
GNU nano 3.2 /etc/dhcpd.conf  
interface eth0  
static ip_address=192.168.10.20/24  
static routers=192.168.10.1  
static domain_name_servers=192.168.10.1  
  
interface eth0  
metric 300  
  
interface wlan0  
metric 200  
[  
  
AG Get Help    AO Write Out  AW Where Is   AK Cut Text   AJ Justify  
AX Exit        AR Read File  AW Replace    AU Uncut Text AT To Spell
```

Figura 3.30: Comandos en terminal de Raspberry Pi para modificar dhcpcd.

```

Node-RED console
File Edit Tabs Help
pi@raspberrypi:~$ sudo node-red-start

Start Node-RED

Once Node-RED has started, point a browser at http://192.168.10.20:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use node-red-stop to stop Node-RED
Use node-red-start to start Node-RED again
Use node-red-log to view the recent log output
Use sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
18 Apr 17:07:15 - [info]
Welcome to Node-RED
=====
18 Apr 17:07:15 - [info] Node-RED version: v0.20.6
18 Apr 17:07:15 - [info] Node.js version: v10.15.2
18 Apr 17:07:15 - [info] Linux 4.19.57-v7+ arm LE
18 Apr 17:07:19 - [info] Loading palette nodes
18 Apr 17:07:26 - [info] Dashboard version 2.20.0 started at /ui
18 Apr 17:07:26 - [info] Settings file : /usr/lib/node_modules/node-red/settings.js
18 Apr 17:07:27 - [info] Context store : 'default' [modules=memory]
18 Apr 17:07:27 - [info] User directory : /home/pi/.node-red
18 Apr 17:07:27 - [warn] Projects disabled : editorTheme.projects.enabled=false
18 Apr 17:07:27 - [info] Flows file : /home/pi/.node-red/flows_raspberrypi.json
18 Apr 17:07:27 - [warn]
-----

```

Figura 3.31: Arranque y ejecución de Node-RED en Raspberry Pi desde el terminal.

Node-RED viene preinstalado con Raspbian y una vez instalado

Node-RED en el Raspberry Pi se debe arrancarlo, tomando en cuenta que es un servicio que se ejecuta localmente en la tarjeta y luego se deberá acceder al mismo desde cualquier navegador. La figura 3.31 muestra los comandos y la respuesta generada al arrancar Node-RED desde el terminal.

Para poder acceder a la ventana de programación de Node-RED se deberá ir a la siguiente dirección desde el navegador: `http://192.168.10.20:1880` o a `http://localhost:1880` y esto nos arrojará hacia la ventana mostrada en la figura 3.32. Aquí se desplegará la ventana de programación donde se empezarán a crear los flujos de programación de conexión con las plataformas IoT y el enlace con el PLC.

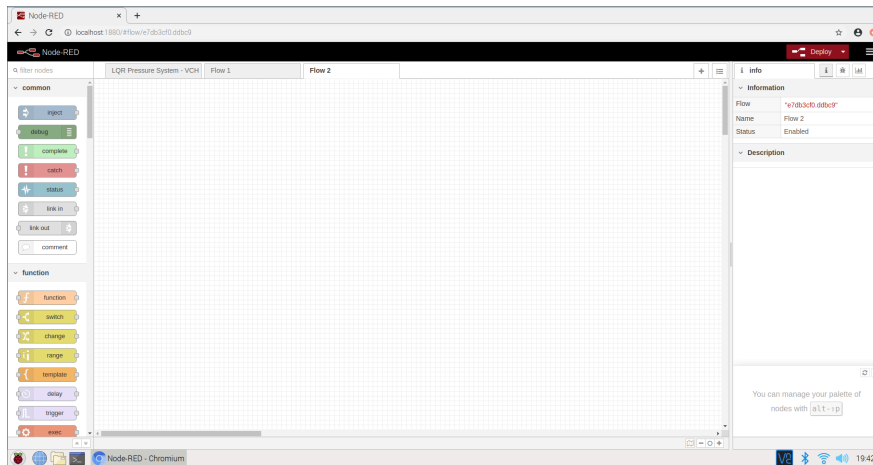


Figura 3.32: Ventana de programación de Node-RED en Raspberry.

3.6.7. Configuración e implementación de enlace entre Node-RED y broker MQTT.

En Node-RED se pueden instalar nodos o librerías creadas, para lo cual se debe instalar el nodo de comunicación MQTT. Esto se lo hará desde la gestión de la paleta de Node-RED y una vez instalado, este permitirá que Node-RED pueda publicar y leer datos desde y hacia el servidor MQTT.

Básicamente, este complemento o nodo se subdivide en dos nodos: un nodo de entrada y otro nodo de salida como se muestra en la figura 3.33.

Estos nodos de entrada y de salida disponen de ciertas propiedades que deben ser configuradas. Es así que los parámetros de configuración se basan en establecer la dirección del broker o servidor MQTT, así como el ID

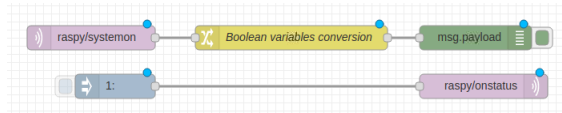


Figura 3.33: Nodos para suscripción y publicación en MQTT.

del cliente. Estos parámetros se indican en la figura 3.34. En este caso, el Raspberry tendrá el nombre de cliente: rasp-client, y la dirección IP es la mencionada anteriormente, la misma que es la dirección IP externa provista por Google Cloud cuando se creó la máquina virtual.

Figura 3.34: Configuración del servidor a usarse en los nodos MQTT en Node-RED.

Una vez configurado el servidor al que el cliente se va a suscribir y sobre el cual va a publicar, se debe configurar el nodo MQTT con el tópico en cuestión, dependiendo de la aplicación. En este caso, como se muestra en la figura 3.35, el tópico es “raspy/systemon”. La configuración es la misma para el nodo de entrada o nodo de salida, siendo importante, definir bien el tópico.

3.6.8. Configuración e implementación de comunicación entre Node-RED y Blynk

En la paleta de Node-RED se instala la librería que permite la conexión segura SSL y la no segura a un servidor local y al servidor de Blynk que se encuentra en la nube. A dicho servidor se accederá usando la dirección `wss://blynk-cloud.com/websockets` para acceder al servidor junto con el token proporcionado al momento de crear una cuenta en Blynk.

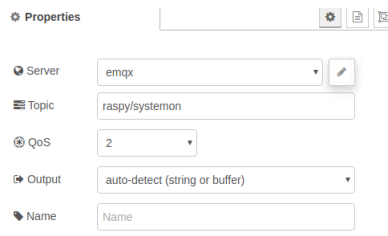


Figura 3.35: Configuración del nodo MQTT en Node-RED.

Se empieza con la creación y parametrización de la conexión con el servidor de Blynk y luego se ingresan los parámetros mostrados en la figura 3.36.

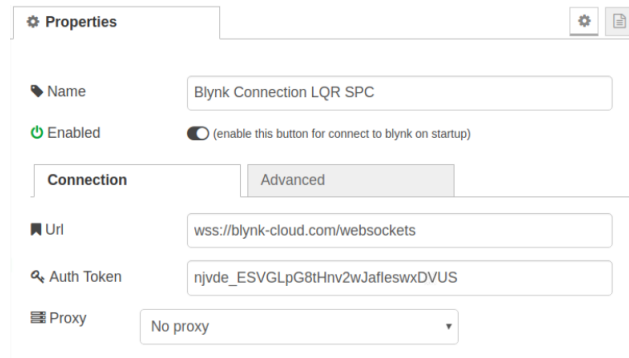


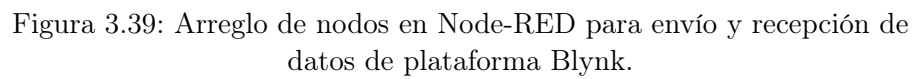
Figura 3.36: Configuración de conexión con servidor de Blynk Cloud.

Es necesario mencionar que los nodos de Blynk tienen eventos de escritura y lectura de datos, por lo cual, se los debe configurar como que se muestra en las figuras 3.37 y 3.38. Primeramente, se selecciona la conexión creada anteriormente. Luego, se selecciona el pin virtual a escribir. Si este pin virtual es diferente, depende de cada elemento usado durante el diseño de la interfaz en Blynk.

La figura 3.39 muestra un arreglo de nodos que lee el valor del botón de la aplicación en Blynk asignado el pin virtual 0 y lo transforma en valores booleanos verdadero o falso según corresponda. Adicionalmente, se tiene un nodo de escritura de datos al pin virtual 4 de la aplicación Blynk, a la entrada tiene valores booleanos enviados de forma manual, los cuales son convertidos a 255 para máxima incandescencia del indicador en la interfaz de Blynk en caso de que el valor a la entrada sea verdadero y 0 para apagado.

Figura 3.37: Configuración de nodo de lectura de Blynk.

Figura 3.38: Configuración de nodo de escritura de Blynk.



3.7. Programación del PLC

3.7.1. Enlace entre PLC Siemens S7-1200 y Raspberry

El enlace entre estos dos dispositivos se lo propuso con el fin de poder enlazar la robustez del PLC y su capacidad de manejo de entradas y salidas aptas para un entorno industrial, con el Raspberry Pi, un dispositivo que permite la conexión IoT, y que de igual manera permite realizar y ejecutar eficientemente algoritmos de control complejos que normalmente no se podrían hacer en el PLC. El diagrama de conexión física entre ambos dispositivos se lo muestra en la figura 3.40, así como también las comunicaciones correspondientes para enlazarse con el broker MQTT.

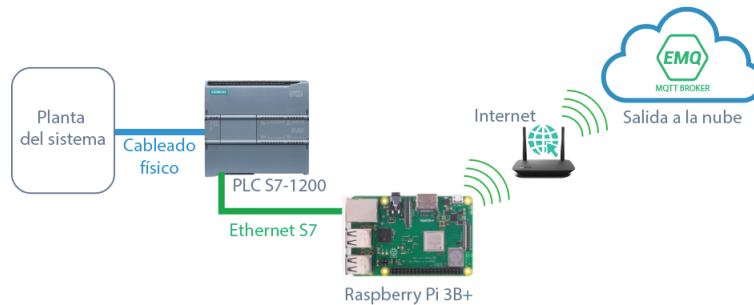


Figura 3.40: Esquema de comunicaciones entre PLC S7-1200, Raspberry y broker MQTT en la nube.

RFC 1006 establece “ISO Transport Service on top of the TCP” el cual es un protocolo de extensión para el protocolo TCP. El protocolo ISO se orienta a paquetes y esta es su principal ventaja. Sin embargo, no dispone de la funcionalidad de routing en las conexiones de red. Por otra parte, el protocolo TCP sí tiene esa funcionalidad. Por ende, se han intentado combinar los puntos positivos de cada protocolo. El protocolo ISO-on-TCP se encuentra en la capa 4 del modelo ISO-OSI, el mismo que usa el puerto 102 para la transmisión de datos.

Además, se pueden intercambiar datos con comunicación S7 entre dispositivos Siemens sobre Ethernet industrial y utilizar las funciones S7, pero Ethernet industrial requiere de los módulos apropiados tipo CP (Communication processor). E inclusive, se pueden usar estos módulos para establecer ISO-transport, ISO-on-TCP, TCP, UDP. Asimismo, usando

PROFINET como el estándar abierto de PROFIBUS, es posible también operar los equipos de entradas y salidas distribuidas en Ethernet industrial [Berger, 2009]. Cabe indicar que S7 es uno de los protocolos principales usados en redes SCADA con equipos Siemens, e incluso otros fabricantes ofrecen equipos compatibles con este protocolo [Kleinmann and Wool, 2014].

La comunicación entre el PLC S7-1200 y la tarjeta Raspberry Pi se la llevará a cabo con la librería nodeS7, la cual permite la comunicación con PLCs S7-300/400/1200/1500 usando el protocolo Ethernet S7 basado en RFC 1006. El protocolo ISO on TCP se emplea para una rápida transferencia de datos y es recomendado para una transferencia de cantidades medianas o grandes de datos y en el caso del PLC S7-1200, se pueden enviar hasta 8 KBytes.

Esta librería establece nodos predeterminados, de entrada y de salida que son configurables, en cuanto a dirección IP del PLC, tipo de comunicación, puerto y nombres de variables con su respectivo registro y tiempo de ciclo, como se representa en la figura 3.41.

Edit s7 out node > Edit s7 endpoint node

Delete Cancel Update

Properties

Connection Variables

Transport Ethernet (ISO-on-TCP)

Address 192.168.10.38 Port 102

Mode Rack/Slot

Rack 0 Slot 1

Cycle time 300 ms

Timeout 1500 ms

Debug Default (command line)

Name S7-1200 AC/DC/RLY

Figura 3.41: Configuración general del nodo S7 en Node-RED.

La figura 3.42 muestra las variables usadas para el enlace del PLC, las cuales fueron puestas en base al programa desarrollado en TIA PORTAL y en base a los registros establecidos de entradas y salidas analógicas. Es así que los creadores de la librería S7 proporcionan un equivalente de los registros y direcciones nativos del PLC y el equivalente que se debe usar al momento de ingresar variables en los nodos S7 junto con el tipo de variable en JS, como se muestra en la figura 3.43.

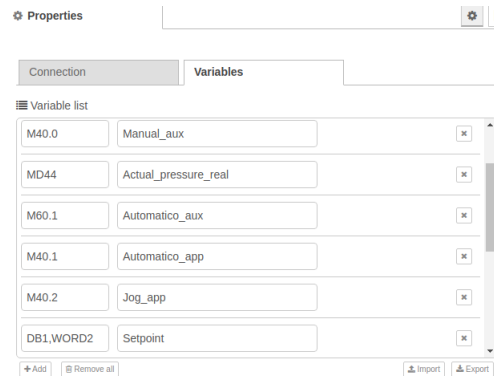


Figura 3.42: Configuración de las variables del nodo S7 en Node-RED.

Address	Step7 equivalent	JS Data type	Description
DB5,X0.1	DB5.DBX0.1	Boolean	Bit 1 of byte 0 of DB 5
DB23,B1 or DB23,BYTE1	DB23.DBB1	Number	Byte 1 (0-255) of DB 23
DB100,C2 or DB100,CHAR2	DB100.DBB2	String	Byte 2 of DB 100 as a Char
DB42,I3 or DB42,INT3	DB42.DBW3	Number	Signed 16-bit number at byte 3 of DB 42
DB57,WORD4	DB57.DBW4	Number	Unsigned 16-bit number at byte 4 of DB 57
DB13,DI5 or DB13,DINT5	DB13.DBD5	Number	Signed 32-bit number at byte 5 of DB 13
DB19,DW6 or DB19,DWORD6	DB19.DBD6	Number	Unsigned 32-bit number at byte 6 of DB 19
DB21,DR7 or DB21,REAL7	DB19.DBD6	Number	Floating point 32-bit number at byte 7 of DB 21
DB2,S7.10*	-	String	String of length 10 starting at byte 7 of DB 2
I1.0 or E1.0	I1.0 or E1.0	Boolean	Bit 0 of byte 1 of input area

Figura 3.43: Equivalentes de variables del nodo S7 en Node-RED.

Una vez establecidas las configuraciones generales que permitirán comunicarse con el PLC Siemens, se debe configurar el nodo, ya sea de entrada o de salida. Se entiende como nodo de entrada al encargado de leer una dirección del PLC, y al nodo de salida como el encargado de escribir ciertos valores sobre una dirección definida del PLC. La figura 3.44 muestra

las casillas de configuración del nodo de entrada/salida. Se debe seleccionar el PLC que se ha configurado anteriormente y luego seleccionar la variable en cuestión y posteriormente, asignar un nombre para fines visuales.

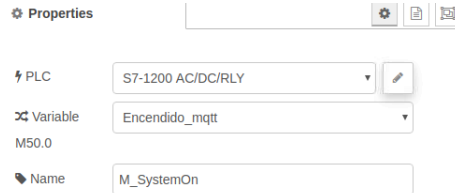


Figura 3.44: Configuración de un nodo de entrada/salida usando S7 en Node-RED.

La figura 3.45 muestra un arreglo de nodos, donde se escribe un valor numérico de 20 al nodo Frequency, mismo que está asignado a la dirección DB1,WORD4 del PLC. Adicionalmente, se lee de igual manera la dirección MD44 del PLC mediante el nodo AnalogInput1, al cual luego se lo convierte a un valor flotante y se lo imprime en la ventana de ejecución.

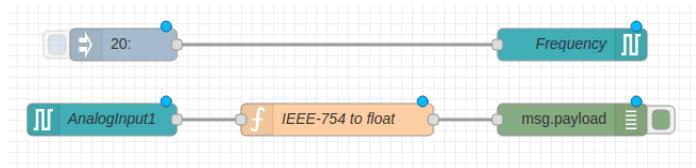


Figura 3.45: Arreglo de nodos para lectura y escritura de una dirección del PLC Siemens.

3.7.2. Algoritmo del PLC

El PLC fue programado en TIA Portal, que es una plataforma de programación de los equipos de Siemens. Dicho programa consiste en un escalamiento de la señal analógica de entrada, la cual primero se la normaliza y luego se la escala al valor requerido en base al rango de presión provisto por el transductor. Posteriormente, esta señal va a la dirección MD44, misma que es leída desde Node-RED. La figura 3.46 muestra el escalamiento de la señal de entrada desde 0 a 145 PSI. Este dato en Node-RED debe ser convertido a flotante para poder trabajar sin problema.

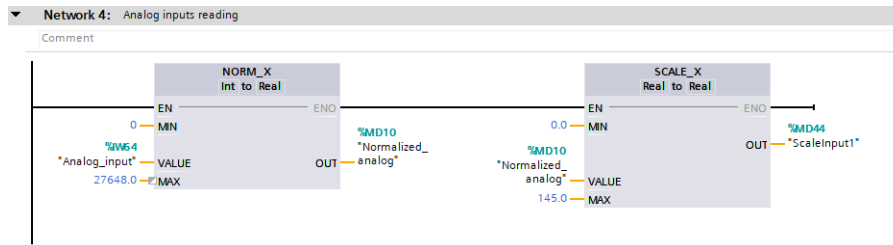


Figura 3.46: Normalizado y escalado de entrada analógica en TIA Portal.

Al tenerse dos modos de operación, un modo automático y un modo manual como se muestra en la figura 3.47, el escalado de la señal analógica de salida deberá ser diferente, dependiendo del modo en el que se esté trabajando. Esta señal dependerá de los datos que el programa en Node-RED envíe a la dirección correspondiente del Data_Block_1 que consta de un arreglo de 10 palabras. El escalado correspondiente se lo puede ver en la figura 3.48.

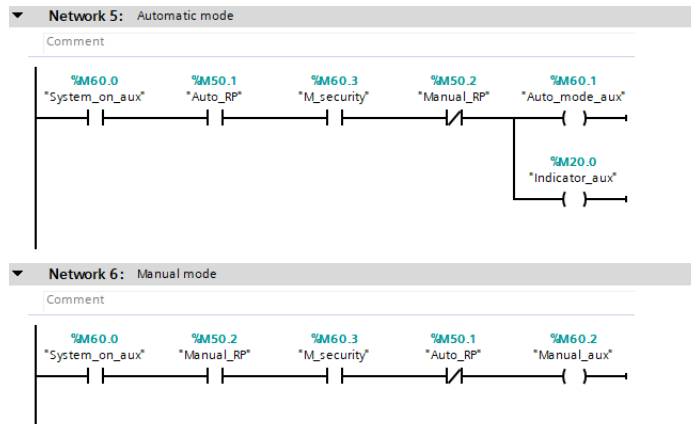


Figura 3.47: Selección de modos de operación del sistema.

A continuación, al obtener una señal analógica escalada entre 0 y 27648, esta es enviada a la dirección de salida analógica que es la QW80 como se muestra en la figura 3.49. Cabe recalcar que los datos movidos a esta dirección son equivalentes a 0 y 10VDC. Esta señal analógica posteriormente irá conectada a la entrada analógica del variador de frecuencia que es el actuador de la planta.

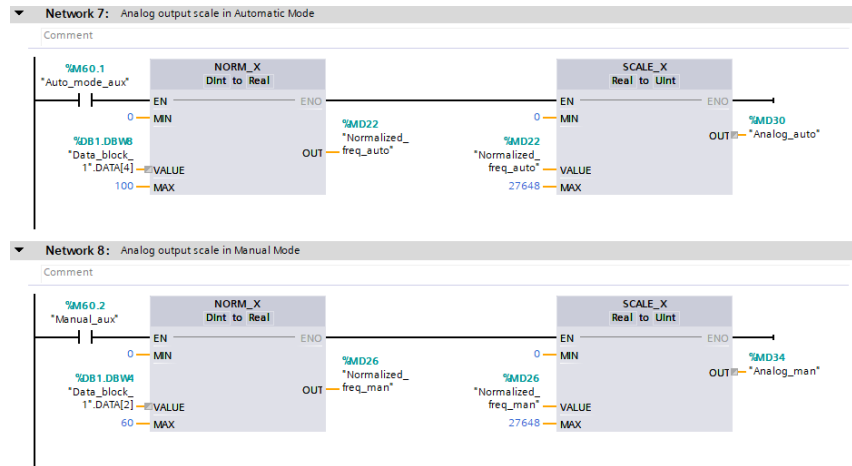


Figura 3.48: Normalizado y escalado de salida analógica en TIA Portal.

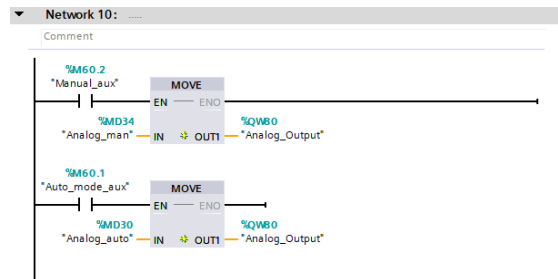


Figura 3.49: Movimiento de datos a dirección de salida analógica del PLC.

Adicionalmente, se asigna la dirección IP del PLC, que en este caso es la 192.168.10.38. Además, se debe permitir la comunicación PUT/GET con otros dispositivos, lo cual permitirá que el Raspberry pueda comunicarse con el PLC sin problemas.

3.8. Diseño del algoritmo de control

El algoritmo de control seleccionado es un control LQR con acción integral. Las siglas LQR se refieren a control óptimo cuadrático, que es una acción de control moderna multi-variable que se caracteriza por su robustez en tiempo continuo y en tiempo discreto [Poveda and Martinez, 2016].

Cuando se trata de una planta no lineal, el cálculo de las constantes de control debe hacerse en base al punto de trabajo seleccionado con anterioridad, dependiendo del grado de no linealidad del sistema y debido a esta no linealidad, el sistema responderá de diferente manera cuando se modifiquen los puntos de trabajo [Poveda and Martinez, 2016]. Por lo tanto, teniendo en cuenta que el funcionamiento de la bomba no es lineal, su curva de trabajo es modificada cuando ésta es controlada con un variador de frecuencia. Por ende, a mayor frecuencia, el sistema incrementará la variable del proceso mucho más rápidamente que cuando se trabaja con frecuencias bajas y por esta razón, se comparará y se diseñará los dos controladores: un LQR y un LQI.

3.8.1. Modelo matemático del sistema

El modelo matemático del sistema fue hallado excitando al actuador a un 50 % de su capacidad. La figura 3.50 muestra el gráfico de entrada versus salida de la planta, donde se puede observar que el sistema se estabiliza en 28 PSI aproximadamente y donde el eje Y de la gráfica de salida es el valor obtenido por el transductor de presión en PSI.

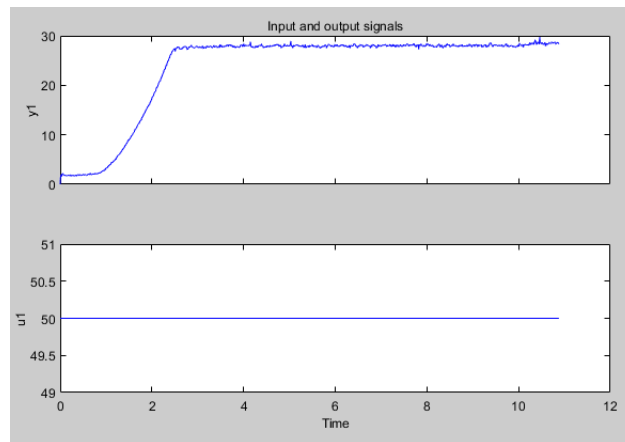


Figura 3.50: Señal de entrada y de salida de la planta ante un escalón de 50 %.

Cabe recalcar también, que el tiempo de muestreo para la obtención de los datos del sistema fue de 10ms. Una vez ingresados estos datos al Matlab, se obtiene una estimación de la función de transferencia, al igual que una estimación de las ecuaciones en espacio de estados, obteniendo la planta

del sistema en tiempo continuo y en tiempo discreto como se muestra a continuación en la figura 3.51, donde se evidencia que la estimación supera un 85 % y por lo tanto, puede considerarse adecuada para el diseño del controlador.

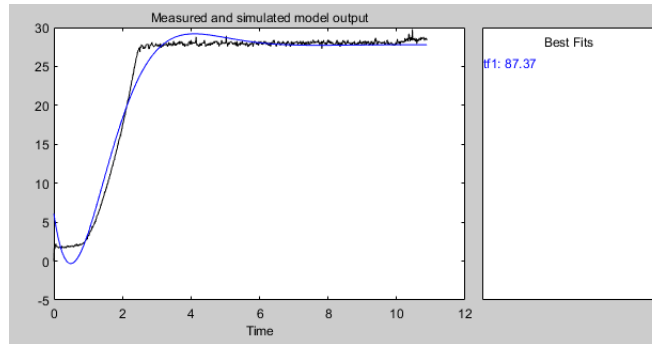


Figura 3.51: Respuesta del modelo estimado versus el sistema original en System Identification de Matlab.

Se considera que el sistema es de segundo orden y es por esto que la respuesta tiene un sobre impulso antes de estabilizarse en el punto de consigna. La figura 3.52 muestra las funciones de transferencia en tiempo continuo y tiempo discreto de 10ms.

```
tf1 =

    From input "u1" to output "y1":
    0.8048
    -----
    s^2 + 1.662 s + 1.448

Name: tf1
Continuous-time identified transfer function.
```

Figura 3.52: Función de transferencia en tiempo continuo del sistema.

Se usa la herramienta “ss” en Matlab para obtener el modelado en espacio de estados de la planta en cuestión a partir de la función de transferencia y se obtienen las siguientes matrices:

$$A = \begin{bmatrix} -1,662 & -1,448 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = [0 \quad 0,8048], D = 0 \quad (3.1)$$

Sin embargo, se requiere obtener el modelo del sistema en tiempo discreto,

por lo cual se usa el comando “c2d” de Matlab, que permite transformar la función de transferencia de tiempo continuo a tiempo discreto. En este caso, se obtiene una función de transferencia en tiempo discreto con un tiempo de muestreo de 10ms, misma que posteriormente se la transforma a espacio de estados y se obtienen las siguientes matrices:

$$A = \begin{bmatrix} 0,9834 & -0,01436 \\ 0,009917 & 0,9999 \end{bmatrix}, B = \begin{bmatrix} 0,009917 \\ 4,972e^{-5} \end{bmatrix}, C = [0 \quad 0,8048], D = 0 \quad (3.2)$$

3.8.2. Controlabilidad y observabilidad del sistema

Un sistema es controlable si converge a una referencia en un tiempo finito. Es decir, que sin importar la respuesta en un momento, este se estabilizará en un punto definido [Poveda and Martinez, 2016].

Para verificar la controlabilidad del sistema se usará el comando “rank” de Matlab, mismo que permite verificar si el sistema es controlable o no, al momento de ejecutar el siguiente código en Matlab, como se muestra en la figura 3.53 la cual arroja un resultado de n=2. Por ende, el sistema es controlable.

```
%check controllability of the system
Pc = [b a*b];
rank(Pc);
```

Figura 3.53: Código para verificar la controlabilidad del sistema.

Además, un proceso es observable cuando los estados del mismo pueden ser obtenidos a partir de las salidas en cada instante de tiempo [Poveda and Martinez, 2016]. El código utilizado se muestra en la figura 3.54 y arroja un resultado de 0, lo cual significa que el sistema es observable.

```
%check observability of the system
Ob = obsv(sysd);
unob = length(a)-rank(Ob);
```

Figura 3.54: Código para verificar la observabilidad del sistema.

3.8.3. Diseño del controlador LQR

Dadas las ecuaciones de un sistema dinámico:

$$\dot{x} = Ax + Bu \quad (3.3)$$

$$y = Cx \quad (3.4)$$

Se tiene una ley de control que debe cumplirse para obtener un controlador LQR en la ecuación 3.5. Sin embargo, también se tiene una función de coste que se muestra en la ecuación 3.6:

$$u(t) = -K \quad (3.5)$$

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (3.6)$$

La solución a este problema de optimización es obtenida de una programación dinámica y el valor de K es dado por la ecuación 3.7.

$$K = R^{-1} B^T S \quad (3.7)$$

La matriz S es la solución a la siguiente ecuación algebraica de Ricatti (ARE) mostrada en la ecuación 3.8, la cual solo tiene solución si el sistema es controlable y observable:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0 \quad (3.8)$$

Cuando se diseña un controlador LQR, las matrices Q y R son cruciales para su diseño usando la técnica LQR que minimiza a J a partir de las variables de control [Poveda and Martinez, 2016]. En este caso, adoptando un control barato y usando un método de prueba y error, se establecen las siguientes matrices:

$$Q = \begin{bmatrix} 1,3 & 0 \\ 0 & 0,4 \end{bmatrix}, \quad R = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (3.9)$$

Ya presentada la parte matemática, se procede a diseñar el controlador usando Matlab y se encuentran los valores de K usando el código mostrado en la figura 3.55. El primer valor de K será almacenado en “state_gain(1,1)” y el segundo valor lo tendrá la variable “state_gain(1,2)”. Estos valores se los muestra en la figura 3.56.

```

%Transfer Function to Space State Equations
ssl = ss(tfl);
[a,b,c,d] = ssdata(ssl);

%Q and R matrix
Q=[1.3 0;
   0 0.4];
R=1;

%LQR estimation in discrete time
K=lqrd(a,b,Q,R,0.3);

state_gain(1,1)=K(1,1);
state_gain(1,2)=K(1,2);

```

Figura 3.55: Código de implementación del controlador LQR.

```

K =

    0.3206    0.0494

```

Figura 3.56: Valores obtenidos de K en diseño de LQR en Matlab.

3.8.4. Diseño del controlador LQI

Básicamente es similar al LQR, sin embargo, en este controlador, la ley de control se expresa en la ecuación 3.10 donde x_i es la salida del integrador. Esta ley de control asegura que la salida “y” siga el comando de referencia o punto de consigna, el número de integradores es igual a la dimensión de la salida “y”.

$$u = -K[x; x_i] \quad (3.10)$$

Esta ley de control adicionalmente minimiza las funciones de coste que se muestran en la ecuación 3.11:

$$J(u) = \sum_{n=0}^{\infty} \{z^T Q_z + u^T R_u + 2z^T N_u\} \quad (3.11)$$

En tiempo discreto, el algoritmo LQI calcula la salida del integrador x_i usando la fórmula de avance de Euler como se indica en la ecuación 3.12, donde T_s es el tiempo de muestreo del sistema.

$$x_i[n+1] = x_i[n] + T_s(r[n] - y[n]) \quad (3.12)$$

Se procede a diseñar el controlador en base a la información provista por Matlab y se usa el complemento “lqi” mismo que acepta el sistema continuo o discreto para el cálculo de las constantes K. En este caso se ingresa el sistema discretizado y se tiene el código mostrado en la figura 3.57, con el cual se obtuvieron los valores de K mostrados en la figura 3.58.

```
%Transfer function to space states
ssl = ss(tfl);
sysd = c2d(ssl,0.3);
[a,b,c,d] = ssdata(sysd);

%check controllability of the system
Pc = [b a*b];
rank(Pc);

%check observability of the system
Ob = obsv(sysd);
unob = length(a)-rank(Ob);

% Q and R matrix
Q=[1.5 0 0;
   0 0.1 0;
   0 0 0.4];
R=2 ;

%LQI estimation
[K,S,e] = lqi(sysd,Q,R);
```

Figura 3.57: Código para el diseño de LQI en Matlab.

```
K =

    0.3722    0.4461   -0.4187
```

Figura 3.58: Constantes K para controlador LQI obtenidas usando Matlab.

3.9. Modelamiento y análisis

Las fases descritas anteriormente están concatenadas por el modelamiento y análisis de las características del sistema con la ayuda de herramientas de simulación.

Para fines de simulación del controlador del sistema se utilizó Simulink de Matlab, donde se colocaron los bloques necesarios para efectuar la implementación y simulación del controlador LQR y del controlador LQI.

3.9.1. Simulación y análisis del controlador LQR

Se efectuó el modelado y simulación del controlador LQR ante un punto de consigna de 28 PSI que fue obtenido cuando se excitó la planta a un 50 % de su capacidad, considerando que debido a la no linealidad del sistema este debería operar en el rango de obtención del sistema. La figura 3.59 muestra el diagrama en Simulink.

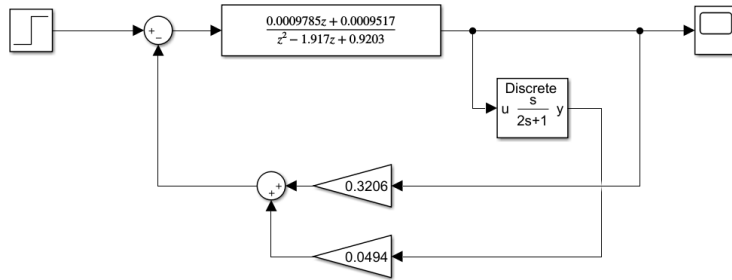


Figura 3.59: Diagrama de bloques de controlador LQR en Simulink.

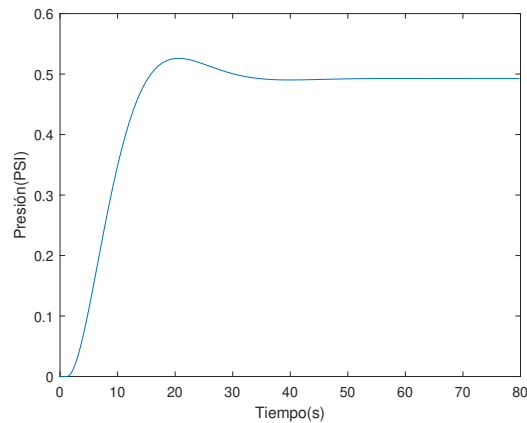


Figura 3.60: Respuesta del sistema con controlador LQR sin compensador a un escalón unitario.

Una vez establecido el diagrama de bloques en Simulink, se procede a enviar un escalón unitario para observar el comportamiento del sistema. Cabe recalcar que el tiempo de muestreo en el bloque de función de transferencia

y en la derivada discreta es de 0.3 segundos, que es el tiempo al que va a trabajar el controlador en la planta real. La figura 3.60 muestra el comportamiento del sistema mostrado en el scope actual.

Debido al diseño utilizado se debe calcular un compensador que será sumado al escalón para que el sistema con control LQR pueda llegar al valor unitario. Dicho compensador se muestra en la ecuación 3.13.

$$Ke = \frac{1}{0,4845} = 2,064 \quad (3.13)$$

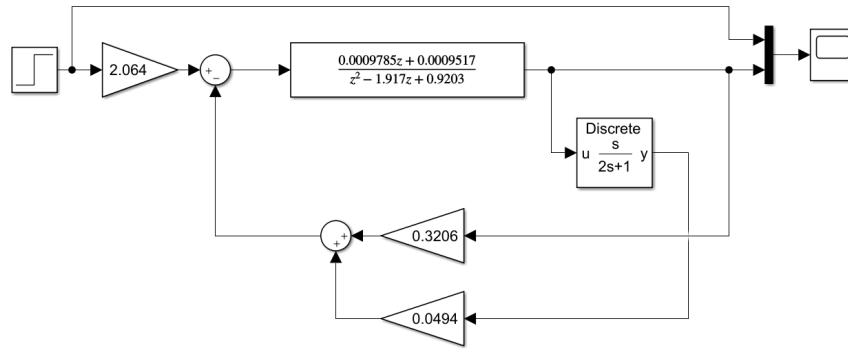


Figura 3.61: Diagrama de bloques en Simulink para algoritmo LQR en tiempo discreto.

Una vez establecida esta ganancia Ke , se la agrega al diagrama de bloques en Simulink a un punto de consigna establecido. En este caso, y considerando el criterio de aplicación del algoritmo LQR, el punto de consigna deberá ser igual o parecido al que se obtuvo del modelo matemático de la planta. En este caso, es de 28 PSI, presión de trabajo establecida. La figura 3.61 muestra el diagrama de bloques en cuestión y la figura 3.62 muestra la respuesta del sistema.

3.9.2. Simulación y análisis del controlador LQI

Se diseña y se implementa este controlador debido a la no linealidad del sistema. Por lo tanto, este algoritmo de control permitirá que se pueda modificar el punto de consigna en varios rangos de operación y el sistema tenga una respuesta adecuada. La figura 3.63 muestra el diagrama de bloques en Simulink del algoritmo LQI y la figura 3.64 la respuesta del sistema.

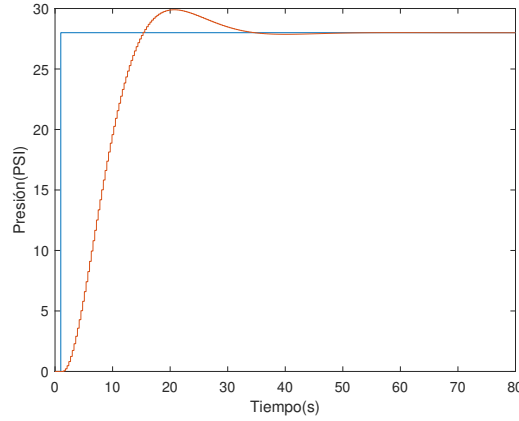


Figura 3.62: Respuesta del sistema con controlador LQR y compensador a 28PSI como punto de consigna.

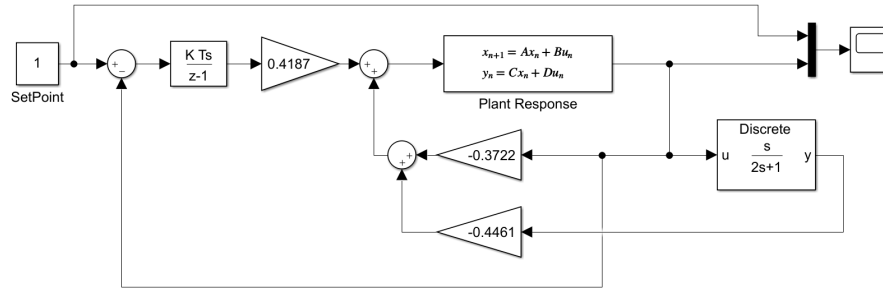


Figura 3.63: Diagrama de bloques en Simulink para algoritmo LQI en tiempo discreto.

3.9.3. Implementación de los controladores LQR y LQI en Node-RED

Los controladores LQR y LQI deberán ser implementados en Node-RED de la Raspberry, puesto que esta es la tarjeta que ejecutará el lazo de control mediante los datos suscritos del broker MQTT y a su vez la salida del controlador la publicará al broker y la enviará a una dirección del PLC. Así, esta señal es acoplada como salida analógica, que a su vez, va a la entrada analógica del variador de frecuencia. La figura 3.65 muestra el arreglo de

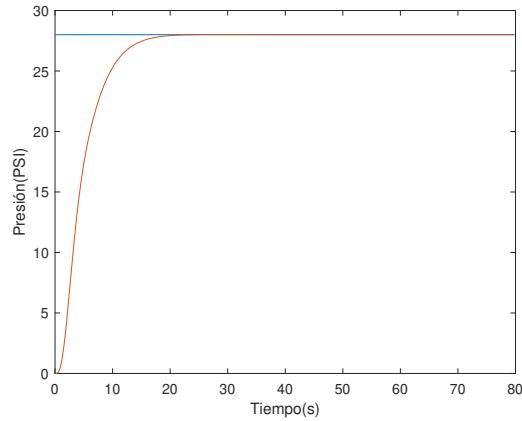


Figura 3.64: Respuesta del sistema con control LQI a 28PSI como punto de consigna.

nodos que leen el valor del transductor de señal y lo convierten a un valor real que entra al bloque de control y posteriormente es enviado nuevamente al PLC. El algoritmo de control, dependiendo del tipo de controlador, es ejecutado en el nodo de función “LQR-LQI Algorithm”. Luego, este bloque debe ser programado en JavaScript. La figura 3.66 muestra parte del código de programación para la ley de control usada en el controlador LQI.

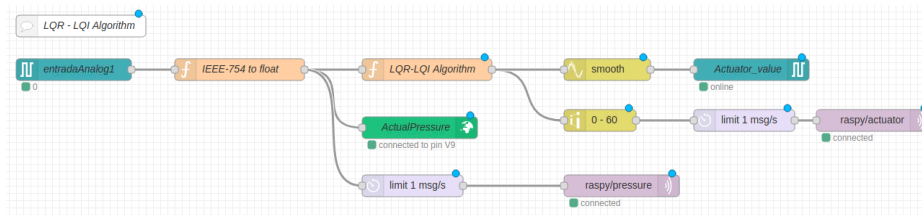



Figura 3.65: Arreglo de nodos en Node-RED para controlador LQI y LQR.

3.10. Integración de sistemas

Los resultados de los dominios específicos serán integrados en un sistema completo para poder analizar las interrelaciones.

Primeramente, se diseñó y construyó el tablero eléctrico de control



```

2 var lastPressureLocal ;
3 const K = [0.3722 , 0.4461 , -0.4187];
4
5 var error;
6 var frequency;
7 var integralErrorLocal;
8
9 var stateVector = [];
10
11 if (flow.get('systemOn')) {
12   //error calculation
13   error = flow.get('reference') - flow.get('pressure');
14
15   //windup
16   if (frequency >= 100 && error > 0) {
17   }
18   else {
19     if (frequency < 0 && error < 0) {
20     }
21     else {
22       //error integral
23       integralErrorLocal = (flow.get('integralError') || 0) + (error * 0.300);
24       flow.set('integralError', integralErrorLocal);
25     }
26   }
27   //pressure
28   stateVector[0] = flow.get('pressure');
29   //
30   lastPressureLocal = flow.get('lastPressure') || flow.get('pressure');
31   //derivative of pressure

```

Figura 3.66: Algoritmo de control en nodo de función en Node-RED usando JavaScript.

mostrado en la figura 3.67, el mismo que se compone de todos los elementos de protección, fuentes de poder, PLC, variador de frecuencia, elementos de mando y control y de igual manera de un Raspberry Pi montado en un soporte para riel din. Este tablero eléctrico dispone de prensa estopas en los laterales por donde se ingresarán los cables del transductor de presión por un lado y por otro, los cables de fuerza del sistema y los cables del motor de la bomba.

La figura 3.68 muestra el resultado de la integración de la parte mecánica e hidráulica y se dispone de la bomba seleccionada con el sistema de tuberías, las válvulas correspondientes, el manómetro y el transductor de presión que será el que provea la retroalimentación al controlador.

Una vez diseñados todos los elementos e ingeniería necesarios para este producto, se obtiene un sistema de presión constante que consta de un tablero eléctrico con sus elementos de control, de un sistema hidráulico, un sistema de monitoreo y control remoto como se muestra en la figura 3.69. Este sistema se lo construyó desde el bosquejo, usando todos los criterios de ingeniería y diseño presentados anteriormente.



Figura 3.67: Tablero eléctrico de control del sistema.



Figura 3.68: Sistema hidráulico resultante.



Figura 3.69: Producto resultante de la integración de sistemas.

3.11. Verificación y validación

El proceso de diseño será revisado continuamente para que el concepto de solución cumpla con los requerimientos deseados. Se realizará la verificación de un adecuado sistema hidráulico, eléctrico, electrónico y de control.

Para la verificación del sistema hidráulico, primero se verificará que la bomba se encuentre cebada. Posteriormente, se verificará el sentido de rotación de la bomba y finalmente se deberá presurizar las líneas hasta 85 PSI y se verificará que no se tengan fugas o roturas de tubería o de los elementos de medición. Para la verificación del sistema eléctrico, se encenderán todos los elementos del sistema como PLC, Raspberry y variador de frecuencia y con el multímetro se realizaron mediciones adecuadas en el sistema.

Se validará el sistema de comunicación remota IoT, tanto en su disponibilidad, como también en la revisión de que los clientes estén siempre conectados al broker MQTT. Esto se validará desde el dashboard del broker. De igual manera, la interfaz IoT en computador y en Smartphone serán validadas usando todos los elementos presentes en ellas y verificando que no hayan demoras o lecturas erróneas de la variable de proceso.

El tipo de controlador implementado será validado en base a la respuesta

del sistema. Si este no tuviese una respuesta adecuada, se deberá calibrar el controlador, modificando las matrices Q y R que afectan la función de costes del controlador. Para la validación del controlador LQI, se deberán realizar cambios en el punto de consigna en un amplio rango de valores, y la respuesta del sistema deberá ser idónea. Adicionalmente, el sistema deberá poder recuperarse ante perturbaciones generadas por la válvula de descarga del sistema.

Capítulo 4

Resultados y Conclusiones

En este capítulo se detallarán los resultados alcanzados en el proyecto desarrollado, así como también las conclusiones a las que se llegaron luego del diseño e implementación de los controladores óptimos y herramientas IoT sobre el sistema de presión constante.

4.1. Resultados y análisis

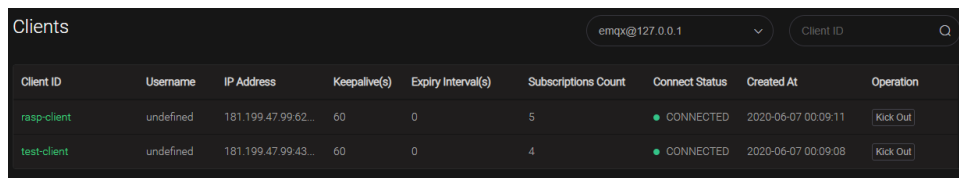
Se elevó la presión del sistema de tuberías hasta 85 PSI y se verificó que no se tengan fugas en las uniones de los elementos tanto al momento de la descarga como al de la succión de la bomba.

Los valores obtenidos de las mediciones de voltaje y corriente son presentados en la tabla 4.1, los mismos que se encuentran dentro de los rangos permisibles y adecuados para un funcionamiento correcto.

Tabla 4.1: Valores de voltaje y corriente obtenidos en el sistema.

Característica	Valor
Voltaje fase-fase	218.5 VAC
Corriente de salida a plena carga del VFD	2.7 Amperios
Corriente máxima total del sistema	4.2 Amperios
Voltaje de salida de fuente 24VDC	23.96 VDC
Voltaje de salida de fuente 5VDC	5.02 VDC

De igual manera al hacer la validación de la comunicación remota entre los clientes con el broker MQTT y al tener un dashboard en el broker, se pudo visualizar que los clientes estén conectados al broker, como se lo muestra en la figura 4.1. Además, se pudo observar que los clientes pueden publicar y suscribirse sin problema alguno.



Client ID	Username	IP Address	Keepalive(s)	Expiry Interval(s)	Subscriptions Count	Connect Status	Created At	Operation
rasp-client	undefined	181.199.47.99.62...	60	0	5	CONNECTED	2020-06-07 00:09:11	Kick Out
test-client	undefined	181.199.47.99.43...	60	0	4	CONNECTED	2020-06-07 00:09:08	Kick Out

Figura 4.1: Verificación de conectividad de broker MQTT con clientes.

Se validó la interfaz gráfica para monitoreo y control remoto durante un tiempo de 30 minutos, tiempo en el cual se cambiaron los puntos de consigna del controlador; se alternó entre modo manual y automático y se compararon los valores de presión en PSI de la interfaz con los del manómetro, cuyo error de medición es de ± 0.5 PSI; se analizaron los tiempos desde que se publican los datos al broker desde el computador, y desde que son suscritos por el Raspberry siendo tiempos menores a 1 segundo, por

lo que la interacción con el sistema es inmediata, sin retardos ni pérdidas de comunicación. Adicionalmente, al tener el broker MQTT instalado en un computador virtual de Google Cloud, la disponibilidad del sistema es continua, gracias a que se usan los mismos servidores de Google que usan redundancia para garantizar su disponibilidad.

El algoritmo de control LQR se lo implementó y se configuró al sistema, a un punto de consigna muy similar, con el que se obtuvo el modelado matemático. Por las razones anteriormente expuestas, la figura 4.2 muestra la interfaz IoT en computador, y en la parte inferior se puede visualizar el histograma con los datos de punto de consigna de 30PSI y de la variable del proceso, establecidos en base a la respuesta del sistema. Adicionalmente, la tabla 4.2 muestra el contraste de ciertos aspectos entre los valores obtenidos en simulación y los del sistema real.

Tabla 4.2: Tabla comparativa de los datos de simulación y de la planta real con controlador LQR.

Descripción	Valor de simulación	Valor real
Tiempo estabilización	30 segundos	18 segundos
Error absoluto	0.05	1
Sobre impulso	SI	SI

Este tipo de controlador no se comporta de una manera óptima cuando se establecen otros puntos de consigna, debido a la no linealidad del sistema como se puede observar en la figura 4.3 ya que el controlador no es capaz de realizar la función de seguimiento al cambio en el punto de consigna debido al tipo de sistema.

Sin embargo, como se muestra en la figura 4.3, al final, cuando se establece un punto de consigna de 29 PSI que es el valor similar o cercano al valor donde la planta se estabilizó cuando se la sometió a un escalón del 50 % del actuador, la respuesta del sistema es correcta.

Por otro lado, de igual manera, se implementó el algoritmo LQI con una respuesta en base a varios puntos de consigna establecidos desde esta misma interfaz y el resultado se lo muestra en el histograma de la figura 4.4. Así, el eje de las abscisas denota al tiempo y el histograma de esta interfaz puede ser modificado para graficar los datos de un tiempo en específico. En este caso, las capturas fueron obtenidas con un tiempo en pantalla de 4 minutos para ambos controladores. Además, para la validación del controlador LQI se establecieron 6 puntos de consigna, 20 PSI, 50 PSI, 60 PSI, 25 PSI, 15

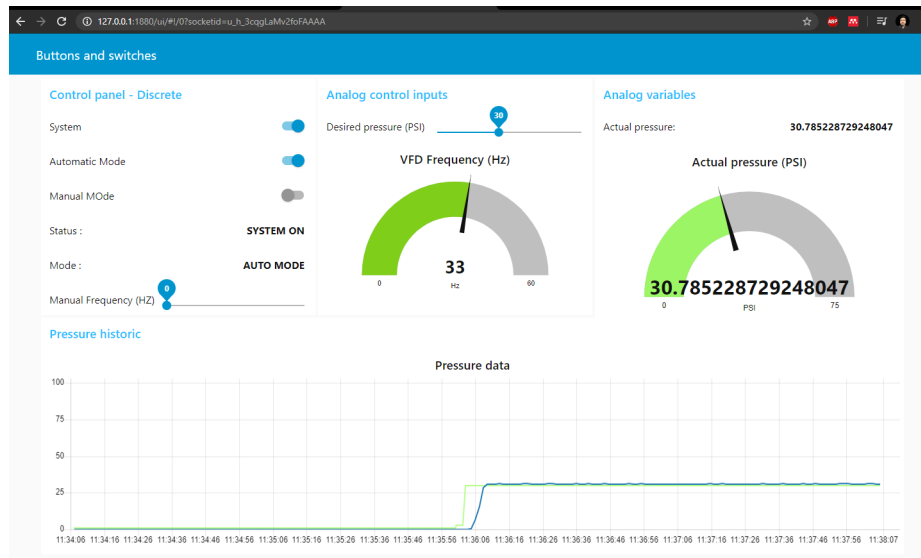


Figura 4.2: Validación del controlador LQR y de la interfaz IoT implementada en el sistema.

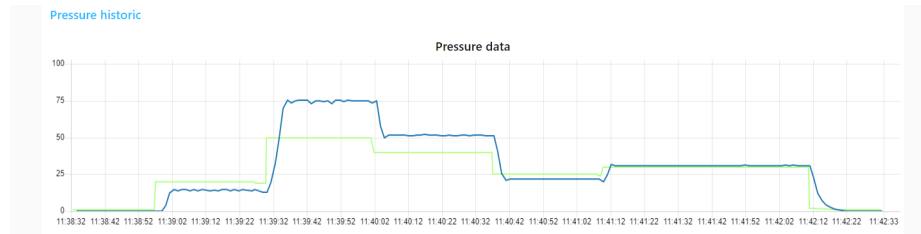


Figura 4.3: Respuesta del sistema con LQR a diferentes puntos de consigna

PSI y 60 PSI. Consecuentemente, se establece claramente que el controlador es capaz de realizar una función de rastreo del punto de consigna definido por el usuario.

Las Figuras 4.5 y 4.6 muestran la respuesta del sistema ante varios puntos de consigna con un controlador LQI. De igual manera se puede observar la interfaz resultante para un computador donde se puede controlar el sistema, configurar, encender, establecer el modo de operación y visualizar variables de proceso. Adicionalmente, en la figura 4.5 se tiene un tiempo en pantalla del histograma de 15 minutos y en la figura 4.6, de 4 minutos.

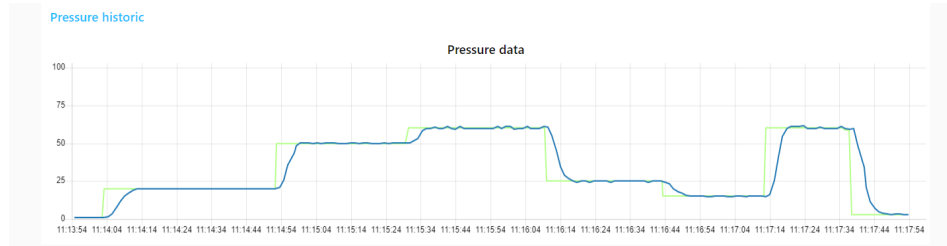


Figura 4.4: Respuesta del sistema con LQI a diferentes puntos de consigna.

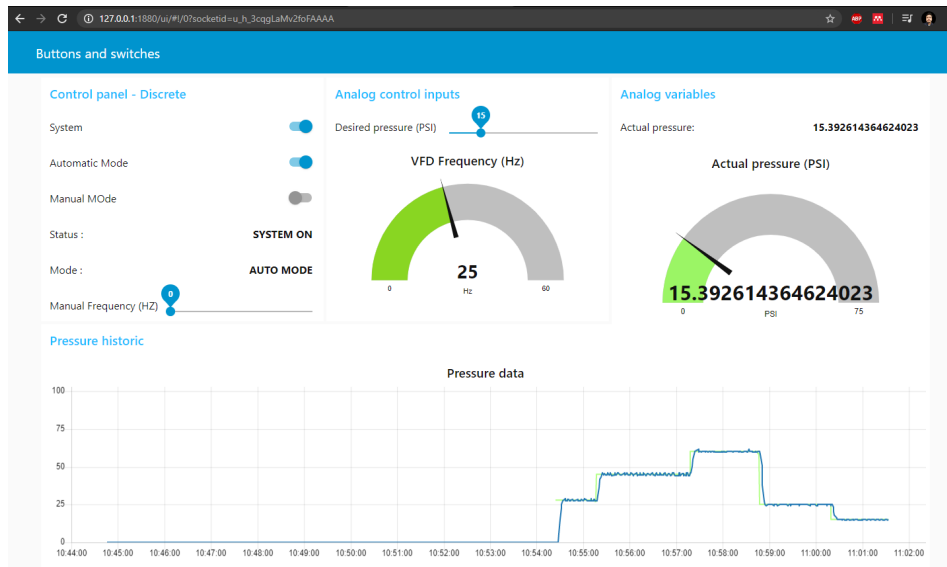


Figura 4.5: Validación 1 del controlador LQI implementado en el sistema.

Se realiza la comparación de tiempo de estabilización, presencia de sobre impulso y error absoluto entre la planta real y la simulación realizada en simulink con la implementación del controlador LQI y se obtienen los datos presentados en la tabla 4.3.

Se analiza y verifica si el controlador es capaz de recuperarse ante perturbaciones, para esto se presenta la figura 4.7 donde se muestra con dos flechas de color azul el momento donde se abrió y cerró la válvula de descarga, el sistema fue sometido a perturbaciones y logró recuperarse en aproximadamente 10 segundos.

Finalmente, se comprueba y analiza el funcionamiento del control y

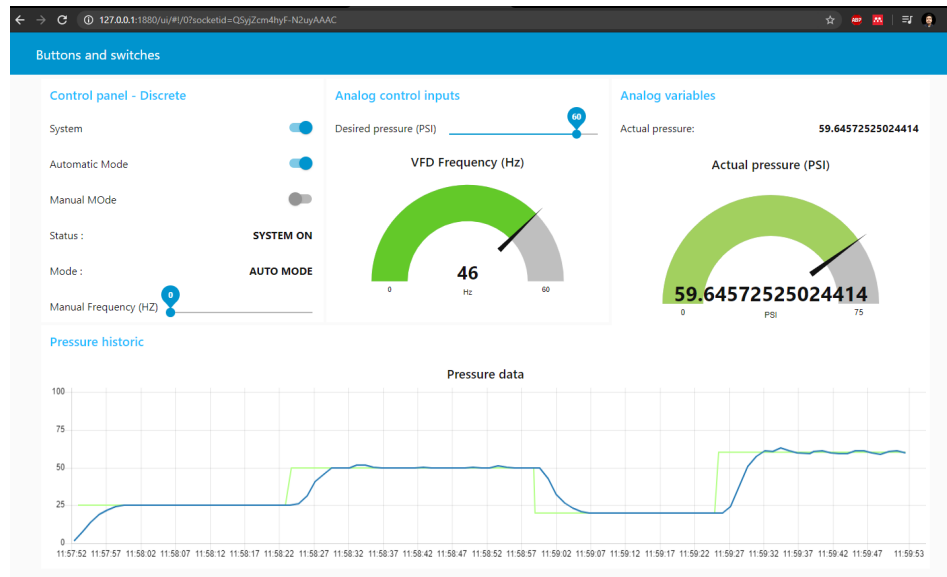


Figura 4.6: Validación 2 del controlador LQI implementado en el sistema.

Tabla 4.3: Tabla comparativa de los datos de simulación y de la planta real con controlador LQI.

Descripción	Valor de simulación	Valor real
Tiempo estabilización	17 segundos	15 segundos
Error absoluto	0.1	1.2
Sobre impulso	NO	NO

monitoreo del sistema desde la aplicación IoT para teléfonos inteligentes en donde los tiempos de reacción son inmediatos ante cambios de puntos de consigna o ante encendido o apagado del sistema. Así, se establece que los dos tipos de interfaz de control y monitoreo remoto, pueden funcionar simultáneamente.

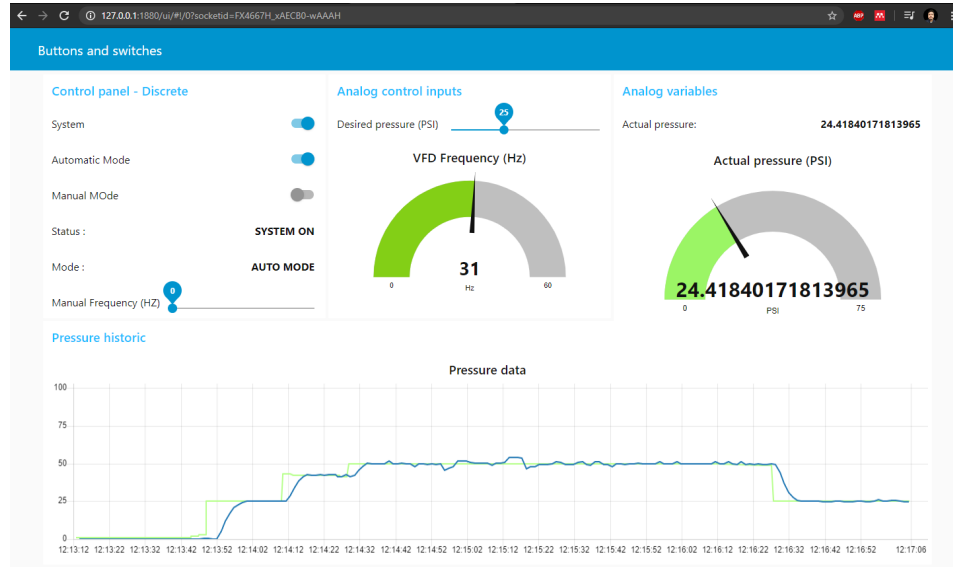


Figura 4.7: Validación 3 del controlador LQI implementado en el sistema ante perturbaciones.

4.2. Conclusiones

Una buena selección de equipos en base a criterios técnicos, garantizan un correcto funcionamiento de cualquier sistema a diseñar, implementar y programar, y son estos equipos los que asegurarán la factibilidad, rentabilidad y durabilidad del proyecto.

Un sistema que es controlable y observable, permite la implementación de controladores óptimos en la Ingeniería de Control, y también asegura que el proceso tenga una respuesta adecuada. En este caso, el controlador LQR demuestra ser un controlador robusto y óptimo para procesos lineales, y para cuando se establece un solo punto de trabajo, sobre el cual se sintoniza el controlador. Sin embargo, cuando se lo aplica a procesos no lineales, su implementación tiene ciertas limitantes, como se lo demostró en este proyecto, ya que el controlador LQR no fue capaz de proveer una adecuada respuesta ante el cambio en el punto de consigna. Por otro lado, el controlador LQI es un controlador lineal cuadrático con acción integral, la misma que permite que este sea apto para sistemas no lineales y pueda realizar una acción de seguimiento ante diferentes puntos de consigna del sistema.

Adicionalmente, el tiempo de estabilización del controlador LQR fue de

8 segundos, comparado con el tiempo de estabilización del controlador LQI que fue de 10 segundos. Sin embargo, estos tiempos dependerán del tipo de ponderación que se seleccionó en el diseño del controlador y en este caso, para ambos controladores, se consideró un coste medio del controlador, de manera que se pueda tener una respuesta equilibrada. Así, estas ponderaciones fueron hechas en base al coste del controlador y a un método de prueba y error para establecer una adecuada respuesta del sistema.

La solución desarrollada en este proyecto para la comunicación IoT entre dispositivos usando un protocolo MQTT para comunicación M2M, permite tener control total sobre los tópicos, clientes, datos, disponibilidad y rentabilidad del sistema. Al tener un computador virtual en los servidores de Google en Latinoamérica, el tiempo de latencia se ve reducido y por ende la comunicación entre los dispositivos es rápida y no se puede apreciar lentitud en la publicación y subscripción de datos. Adicionalmente, al no tener instalado el broker MQTT en un computador físico, se puede garantizar su disponibilidad permanente.

Por otra parte, la plataforma Blynk, al ser de uso comercial pagado, en su versión sencilla dispuso de los widgets mínimos necesarios para realizar esta aplicación. Sin embargo, no provee acceso al broker, ni se pueden visualizar ni controlar los tópicos o clientes usados para la comunicación MQTT, al contrario de la solución desarrollada en el proyecto usando Node-RED.

La instalación de Node-RED en un computador con Windows, Linux o MacOS dedica ciertas habilidades que las tendría el instalador, sin embargo, para aplicar este tipo de sistemas a nivel industrial y comercial, se recomendaría usar un computador dedicado para este fin, caso contrario cuando el usuario esté fuera de las instalaciones, se debería usar la aplicación móvil.

Además, el sistema desarrollado en el presente trabajo, es escalable y se puede ajustar a los diferentes requerimientos del cliente, partiendo desde la comunicación entre Raspberry y los PLC Siemens S7-1200, S7-300 y S7-1500, lo cual facilita su implementación y no se requiere de cableados adicionales, sino solamente de un cable de red o un switch que permita que el Raspberry acceda a las direcciones de memoria del PLC. Así, este sistema puede aplicarse para cualquier tipo de control requerido en la industria, en la cual, durante muchos años, se han usado netamente controladores PID. Por otra parte, inclusive, la inversión requerida para instalar un sistema de control óptimo con monitoreo y control remoto se ve reducida, puesto que los costos de implementación de la plataforma IoT dependen básicamente del costo del Raspberry, de su tablero de control, costos de ingeniería y de la mínima mensualidad del computador virtual en Google Cloud que permitirá

tener un broker MQTT con una variedad de clientes y tópicos.

Finalmente, el presente trabajo brinda un aporte para la sinergia entre las ramas técnicas y teóricas de la Ingeniería en Control, puesto que implementa dos controladores óptimos en una planta real escalable muy común a nivel industrial, donde se puede validar y verificar el diseño realizado con software de simulación con la respuesta real del sistema, con un actuador con limitantes en un sistema que no es lineal y por ende no es el ideal. Sin embargo, con un conocimiento de las bases teóricas de control, se puede llegar a obtener una respuesta óptima del sistema.

Bibliografía

- A. M. Abdurraqueeb, K. Alsharabi, M. Aljalal, and W. Ko. Design state space feedback and optimal LQR controllers for load frequency in hydraulic power system. *2019 8th International Conference on Modeling Simulation and Applied Optimization, ICMSAO 2019*, pages 1–5, 2019. doi: 10.1109/ICMSAO.2019.8880378.
- M. Al-dhaifalla. Design of Intelligent Fuzzy Fractional -Order PID Controller for Pressure Control Application. pages 525–530, 2017.
- M. Álvarez. *Convertidores de frecuencia, controladores de motores y SSR*. MARCOMBO, S.A., 2000. URL <https://books.google.com.ec/books?id=iWP2zQuT0l8C{&}printsec=frontcover{#}v=onepage{&}q{&}f=false>.
- I. B. Aris, R. K. Z. Sahbusdin, and A. F. M. Amin. Impacts of IoT and big data to automotive industry. *2015 10th Asian Control Conference: Emerging Control Techniques for a Sustainable World, ASCC 2015*, pages 1–5, 2015. doi: 10.1109/ASCC.2015.7244878.
- Z. B. Babovic, J. Protic, and V. Milutinovic. Web Performance Evaluation for Internet of Things Applications. *IEEE Access*, 4:6974–6992, 2016. ISSN 21693536. doi: 10.1109/ACCESS.2016.2615181.
- L. Bassi. Industry 4.0: Hope, hype or revolution? *RTSI 2017 - IEEE 3rd International Forum on Research and Technologies for Society and Industry, Conference Proceedings*, 2017. doi: 10.1109/RTSI.2017.8065927.
- H. Berger. *Automating with SIMATIC : controllers, software, programming, data communication, operator control and process monitoring*. Publicis Pub, 2009. ISBN 3895783331.
- J. Bharali and M. Buragohain. Design and performance analysis of Fuzzy LQR; Fuzzy PID and LQR controller for active suspension system using 3

- Degree of Freedom quarter car model. *1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems, ICPEICES 2016*, pages 1–6, 2017. doi: 10.1109/ICPEICES.2016.7853369.
- S. Chanthakit and C. Rattanapoka. Mqtt based air quality monitoring system using node MCU and node-red. *Proceeding of 2018 7th ICT International Student Project Conference, ICT-ISPC 2018*, pages 1–5, 2018. doi: 10.1109/ICT-ISPC.2018.8523891.
- K. Ferencz and J. Domokos. Rapid Prototyping of IoT Applications for the Industry. *2020 22nd IEEE International Conference on Automation, Quality and Testing, Robotics - THETA, AQTR 2020 - Proceedings*, 2020. doi: 10.1109/AQTR49680.2020.9129934.
- J. Gausemeier and S. Moehringer. VDI 2206- A New Guideline for the Design of Mechatronic Systems. *IFAC Proceedings Volumes*, 35(2):785–790, 2002. ISSN 14746670. doi: 10.1016/s1474-6670(17)34035-1. URL [http://dx.doi.org/10.1016/S1474-6670\(17\)34035-1](http://dx.doi.org/10.1016/S1474-6670(17)34035-1).
- W. Q. He, M. K. Cai, Y. B. Wang, and X. J. Wang. Automatic water supply control system of graded constant pressure by variable frequency speed and its application to pipeline irrigation. *Proceedings - 2010 2nd WRI Global Congress on Intelligent Systems, GCIS 2010*, 1:385–388, 2010. doi: 10.1109/GCIS.2010.81.
- Y. Itani, M. Soliman, and M. Kahil. Recovering energy by hydro-turbines application in water transmission pipelines: A case study west of saudi arabia. *Energy*, 211, 2020. doi: 10.1016/j.energy.2020.118613. URL <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85090243743&doi=10.1016%2fj.energy.2020.118613&partnerID=40&md5=99102ccc8246b223434efff24c15cc3e>. cited By 0.
- H. Kato, H. Fujimoto, and K. Yamashina. Operational improvement of main pumps for energy-saving in wastewater treatment plants. *Water (Switzerland)*, 11(12):1–13, 2019. ISSN 20734441. doi: 10.3390/W11122438.
- A. Kleinmann and A. Wool. Accurate Modeling of the Siemens S7 SCADA Protocol for Intrusion Detection and Digital Forensics. *Journal of Digital Forensics, Security and Law*, 9(2), 2014. ISSN 1558-7223. doi: 10.15394/jdfsl.2014.1169.

- V. Lampkin, W. T. Leong, L. Olivera, S. Rawat, N. Subrahmanyam, and R. Xiang. Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry. *IBM Redbooks*, page 270, 2012. URL <http://books.google.com/books?hl=en&lr={&}id=F{ }HHAgAAQBAJ{&}oi=fnd{&}pg=PP1{&}dq=Building+Smarter+Planet+Solutions+with+MQTT+and+IBM+WebSphere+MQ+Telemetry{&}ots=2CE3veSYOD{&}sig=CR-rE8p9kZqMzAno6nzLy7G29qg>.
- M. Lekić and G. Gardašević. IoT sensor integration to Node-RED platform. *2018 17th International Symposium on INFOTEH-JAHORINA, INFOTEH 2018 - Proceedings*, 2018-Janua(March):1–5, 2018. doi: 10.1109/INFOTEH.2018.8345544.
- S. Li and J. Liu. Design and application of auto-constant water supply system. *Proceedings of the 2009 IEEE International Conference on Automation and Logistics, ICAL 2009*, (August):623–627, 2009. doi: 10.1109/ICAL.2009.5262847.
- C. Maheswari, E. B. Priyanka, S. Thangavel, S. V. Vignesh, and C. Poongodi. Multiple regression analysis for the prediction of extraction efficiency in mining industry with industrial IoT. *Production Engineering*, (0123456789), 2020. ISSN 18637353. doi: 10.1007/s11740-020-00970-z. URL <https://doi.org/10.1007/s11740-020-00970-z>.
- L. I. Minchala, J. Peralta, P. Mata-Quevedo, and J. Rojas. An approach to industrial automation based on low-cost embedded platforms and open software. *Applied Sciences (Switzerland)*, 10(14), 2020. ISSN 20763417. doi: 10.3390/app10144696.
- A. Mohebbi, S. Achiche, and L. Baron. Integrated and concurrent detailed design of a mechatronic quadrotor system using a fuzzy-based particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 82(May 2018):192–206, 2019. ISSN 09521976. doi: 10.1016/j.engappai.2019.03.025.
- B. Nesbitt. *Handbook of Pumps and Pumping : Pumping Manual International*. Elsevier Science & Technology, Jordan Hill, UNITED KINGDOM, 2006. ISBN 9780080549217. URL <http://ebookcentral.proquest.com/lib/upsal/detail.action?docID=330202>.
- S. Ozana, M. Pies, Z. Slanina, and R. Hajovsky. Design and implementation of LQR controller for inverted pendulum by use of REX control system.

- International Conference on Control, Automation and Systems*, pages 343–347, 2012. ISSN 15987833.
- P. Panwar, S. Mukhopadhyay, T. U. Bhatt, and A. P. Tiwari. Design of Decentralized PI and LQR Controllers for Level and Pressure Regulation of Steam Drum of Advanced Heavy Water Reactor. *2018 Indian Control Conference (ICC)*, i(Icc):258–263, 2018. doi: 10.1109/INDIANCC.2018.8307988.
- D. H. Park, H. C. Bang, C. S. Pyo, and S. J. Kang. Semantic open IoT service platform technology. *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*, pages 85–88, 2014. doi: 10.1109/WF-IoT.2014.6803125.
- Y. Peng, P. Liu, and T. Fu. Performance analysis of edge-PLCs enabled industrial Internet of things. *Peer-to-Peer Networking and Applications*, 13(5):1830–1838, 2020. ISSN 19366450. doi: 10.1007/s12083-020-00934-1.
- R. Petrasch and R. Hentschke. Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method. *2016 13th International Joint Conference on Computer Science and Software Engineering, JCSSE 2016*, (Cc):1–5, 2016. doi: 10.1109/JCSSE.2016.7748885.
- T. Poveda and J. Martinez. Diseño e implementación de un Control Óptimo LQR con la tarjeta Raspberry Pi. page 104. 2016.
- E. B. Priyanka, C. Maheswari, S. Thangavel, and M. P. Bala. Integrating IoT with LQR-PID controller for online surveillance and control of flow and pressure in fluid transportation system. *Journal of Industrial Information Integration*, 17(January):100127, 2020. ISSN 2452414X. doi: 10.1016/j.jii.2020.100127. URL <https://doi.org/10.1016/j.jii.2020.100127>.
- P. V. Rhyn. Increasing Water Pump Station Throughput by Introducing VFD-based IE4 Class Synchronous Reluctance Motors with Improved Pump Control. *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, pages 1–6, 2018.
- A. Shankar, S. Umashankar, and S. Paramasivam. Investigations on Performance Evaluation of VFD fed PMSM using DTC Control Strategies for Pumping Applications. *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, pages 1–8, 2017. doi: 10.1109/IPACT.2017.8244995.

- Y. Yu, Y. Wang, C. Xie, X. Zhang, and W. Jiang. A proposed approach to mechatronics design education: Integrating design methodology, simulation with projects. *Mechatronics*, 23(8):942–948, 2013. ISSN 09574158. doi: 10.1016/j.mechatronics.2012.10.002.
- L. Zhang and X. Zhuan. Optimization on the VFDs’ Operation for Pump Units. *Water Resources Management*, 33(1):355–368, 2019. ISSN 15731650. doi: 10.1007/s11269-018-2106-2.